Research Article

HLRF-BFGS-Based Algorithm for Inverse Reliability Analysis

Rakul Bharatwaj Ramesh, Olivia Mirza, and Won-Hee Kang

School of Computing Engineering and Mathematics, Western Sydney University, Penrith, NSW 2747, Australia

Correspondence should be addressed to Won-Hee Kang; W.Kang@westernsydney.edu.au

Received 10 March 2017; Accepted 7 June 2017; Published 17 July 2017

Academic Editor: Roman Lewandowski

Copyright © 2017 Rakul Bharatwaj Ramesh et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This study proposes an algorithm to solve inverse reliability problems with a single unknown parameter. The proposed algorithm is based on an existing algorithm, the inverse first-order reliability method (inverse-FORM), which uses the Hasofer Lind Rackwitz Fiessler (HLRF) algorithm. The initial algorithm analyzed in this study was developed by modifying the HLRF algorithm in inverse-FORM using the Broyden-Fletcher-Goldarb-Shanno (BFGS) update formula completely. Based on numerical experiments, this modification was found to be more efficient than inverse-FORM when applied to most of the limit state functions considered in this study, as it requires comparatively a smaller number of iterations to arrive at the solution. However, to achieve this higher computational efficiency, this modified algorithm sometimes compromised the accuracy of the final solution. To overcome this drawback, a hybrid method by using both the algorithms, original HLRF algorithm and the modified algorithm with BFGS update formula, is proposed. This hybrid algorithm achieves better computational efficiency, compared to inverse-FORM, without compromising the accuracy of the final solution. Comparative numerical examples are provided to demonstrate the improved performance of this hybrid algorithm over that of inverse-FORM in terms of accuracy and efficiency.

1. Introduction

The reliability of structural members is a basic criterion of their design [1–6]. One of the widely used nonsimulation based methods for reliability estimation is the first-order reliability method (FORM) [7]. It uses a linear approximation of the limit state function. Nonnormal variables in their original space are generally mapped to the standard normal space. This is carried out to obtain the most probable point of failure, also called the design point. In the standard normal space, $\mathbf{u}$ is defined as a column vector of standard normal variables, and $G(\mathbf{u})$ is defined as the limit state function. Note that all the vectors mentioned in this paper are column vectors. The design point is defined as the point on the limit state boundary ($G(\mathbf{u}) = 0$), which is closest to the origin.

In first-order reliability method (FORM), reliability index β is estimated as the magnitude of the design point vector in the standard normal space; that is, $\beta = \|\mathbf{u}\|$ when $\mathbf{u}$ satisfies the following optimization condition:

$$\begin{aligned} \min \quad & \left(\frac{1}{2} \mathbf{u}^T \mathbf{u} \right) \\ \text{subject to} \quad & G(\mathbf{u}) = 0. \end{aligned} \quad (1)$$

In an inverse reliability problem, a target reliability index β_t is generally given, and an unknown parameter represented by θ in a limit state function is inversely calculated. In practical cases, θ can represent any parameter in a limit state function that must be chosen to obtain a specific level of reliability. The value of θ needs to be determined at the design point that satisfies $\|\mathbf{u}\| = \beta_t$ and $G(\mathbf{u}, \theta) = 0$. The inverse reliability problem is defined by the following equation set:

$$\|\mathbf{u}\| - \beta_t = 0, \quad (2a)$$

$$\mathbf{u} + \frac{\|\mathbf{u}\|}{\|\nabla_{\mathbf{u}} G(\mathbf{u}, \theta)\|} \nabla_{\mathbf{u}} G(\mathbf{u}, \theta) = 0, \quad (2b)$$

$$G(\mathbf{u}, \theta) = 0, \quad (2c)$$

where $\nabla_{\mathbf{u}}$ is the gradient operator with respect to $\mathbf{u}$. Simply, the value of θ can be found by a *trial and error* method using a forward reliability procedure (e.g., FORM) and then interpolating the value of the unknown parameter for a specific reliability level. However, this method is tedious requiring computational costs [8], and inverse reliability algorithms are far more advantageous than direct reliability measures [9].

The solution for an inverse reliability problem can be obtained using several methods. A reliability contour method was developed by Winterstein et al. [10] and applied to problems in offshore structure problems. As an extension of this method to general limit states, Der Kiureghian et al. [11] developed an algorithm, inverse-FORM, which was based on the Hasofer Lind Rackwitz Fiessler (HLRF) algorithm [12–14]. This inverse reliability method was used by other researchers for various applications including extreme and nominal loads on wind turbines [15, 16] and critical earthquake loads [17]. This HLRF-based method was found to have a fast convergence as demonstrated by the numerical examples considered by Der Kiureghian et al. [11]. However, when the limit state function is highly nonlinear, this method may converge slowly [18].

In solving forward reliability analyses, sequential quadratic programming techniques such as the BFGS (Broyden-Fletcher-Goldfarb-Shanno) formula have been used in the literature [19, 20]. BFGS-based algorithms were found to be a cheaper alternative to HLRF-based algorithms in solving a certain set of problems [21, 22]. These algorithms were proved to be more robust than HLRF-based algorithms in solving forward reliability problems [14]. In addition, they were also found to be more efficient in solving large scale constraint-based reliability problems [23, 24]. Although this method has been used in solving forward reliability problems, this study is the first attempt to extend the applications of the BFGS formula to inverse reliability analysis.

In this paper, a hybrid inverse reliability algorithm based on the BFGS update formula is proposed. In inverse-FORM, the HLRF-based algorithm, the Hessian of the Lagrangian is approximated by the identity matrix for every iteration as shown by Perićaro et al. [18]. However, the proposed hybrid method uses the BFGS update formula in some iterations to calculate the Hessian of the Lagrangian.

Initially, when developing this hybrid algorithm, the idea of using the BFGS update formula for every iteration was tried as an intermediate algorithm. Although this modification improved the efficiency of solving the numerical examples, it affected the accuracy of the final solution. Hence, further modifications were made to arrive at the hybrid algorithm that uses the identity matrix in few iterations and the BFGS update formula in other iterations for approximating the Hessian of the Lagrangian. This hybrid algorithm achieves better efficiency without compromising the accuracy of the final solution. In the following sections, the mathematical formulations of inverse-FORM and the proposed hybrid algorithm are provided for inverse reliability analysis. The method proposed in this paper can be used to analyze the reliability of a component in a system.

2. Inverse-FORM Algorithm

The inverse-FORM algorithm was proposed by Der Kiureghian et al. [11] to solve (2a), (2b), and (2c). They proposed using a merit function f to induce convergence to the solution and a step direction vector $\mathbf{d}$, which represents a direction vector towards the next searching point in the subsequent iteration. The merit function f is composed of two subfunctions, f_1 and f_2 . f_1 ensures the convergence of the reliability index at the search point (β) to the target reliability index (β_t); f_2 ensures that the search point converges to a point on the limit state boundary. The step direction vector ($\mathbf{d} = [\mathbf{d}_u^T d_\theta]^T$) also has two parts: $\mathbf{d}_u$ and d_θ . $\mathbf{d}_u$ represents a direction vector towards the next values of the design variables in the subsequent iteration ($\mathbf{u}_{i+1}$). d_θ represents a direction constant towards the next value of the unknown parameter in the subsequent iteration (θ_{i+1}).

In the inverse-FORM algorithm, the first starting values for $\mathbf{u}$ and θ are selected as $\mathbf{u}_0$ and θ_0 , respectively, and they are updated using the following equation until converged:

$$\begin{bmatrix} \mathbf{u}_{i+1} \\ \theta_{i+1} \end{bmatrix} = \begin{bmatrix} \mathbf{u}_i \\ \theta_i \end{bmatrix} + \lambda_i \mathbf{d}, \quad (3)$$

where λ_i is the step size for the i th iteration; $\mathbf{d}$ is the step direction vector; $\mathbf{u}_i$ and $\mathbf{u}_{i+1}$ are the values of $\mathbf{u}$ in the i th and $(i + 1)$ th iterations, respectively; θ_i and θ_{i+1} are the values of θ in the i th and $(i + 1)$ th iterations, respectively.

The step size λ_i in (3) is determined satisfying the following condition:

$$f(\mathbf{u}_{i+1}, \theta_{i+1}) < f(\mathbf{u}_i, \theta_i), \quad (4)$$

where f is the merit function given by the following equation:

$$f(\mathbf{u}, \theta) = f_1(\mathbf{u}, \theta) + f_2(\mathbf{u}, \theta), \quad (5)$$

where f_1 and f_2 are given by the following equations:

$$f_1 = \frac{1}{2} \left\| \mathbf{u} - \left[\frac{\nabla_{\mathbf{u}} G(\mathbf{u}, \theta)^T \mathbf{u}}{\nabla_{\mathbf{u}} G(\mathbf{u}, \theta)^T \nabla_{\mathbf{u}} G(\mathbf{u}, \theta)} \right] \nabla_{\mathbf{u}} G(\mathbf{u}, \theta) \right\|^2 + \frac{1}{2} c_1 G(\mathbf{u}, \theta)^2, \quad (6)$$

$$f_2 = \frac{1}{2} (\|\mathbf{u}\| - \beta_t)^2, \quad (7)$$

where c_1 is given by the following equation:

$$c_1 = \frac{\max \left[\left\| \mathbf{u} - \left[\frac{\nabla_{\mathbf{u}} G(\mathbf{u}, \theta)^T \mathbf{u}}{\nabla_{\mathbf{u}} G(\mathbf{u}, \theta)^T \nabla_{\mathbf{u}} G(\mathbf{u}, \theta)} \right] \nabla_{\mathbf{u}} G(\mathbf{u}, \theta) \right\|^2, 0.001^2 \right]}{\max [G(\mathbf{u}, \theta)^2, 0.001^2]}. \quad (8)$$

The step direction vector ($\mathbf{d}$) in (3) is calculated as follows:

$$\mathbf{d} = a_1 \mathbf{d}_1 + a_2 \mathbf{d}_2, \quad (9)$$

where a_1 , a_2 , $\mathbf{d}_1$, and $\mathbf{d}_2$ are calculated as follows:

$$a_1 = \frac{f_1(\mathbf{u}, \theta)}{f_1(\mathbf{u}, \theta) + f_2(\mathbf{u}, \theta)}, \quad (10)$$

$$a_2 = \frac{f_2(\mathbf{u}, \theta)}{f_1(\mathbf{u}, \theta) + f_2(\mathbf{u}, \theta)}, \quad (11)$$

$$\mathbf{d}_1 = \begin{bmatrix} \mathbf{d}_{1u} \\ d_{1\theta} \end{bmatrix}, \quad (12)$$

$$\mathbf{d}_2 = \begin{bmatrix} \mathbf{d}_{2u} \\ d_{2\theta} \end{bmatrix}, \quad (13)$$

where $\mathbf{d}_{1u}$, $d_{1\theta}$, $\mathbf{d}_{2u}$, and $d_{2\theta}$ are calculated as follows:

$$\mathbf{d}_{1u} = \left[\frac{\nabla_{\mathbf{u}} G(\mathbf{u}, \theta)^T \mathbf{u} - G(\mathbf{u}, \theta)}{\|\nabla_{\mathbf{u}} G(\mathbf{u}, \theta)\|^2} \right] \nabla_{\mathbf{u}} G(\mathbf{u}, \theta) - \mathbf{u}, \quad (14)$$

$$d_{1\theta} = 0, \quad (15)$$

$$\mathbf{d}_{2u} = -\beta_t \frac{\nabla_{\mathbf{u}} G(\mathbf{u}, \theta)}{\|\nabla_{\mathbf{u}} G(\mathbf{u}, \theta)\|} - \mathbf{u}, \quad (16)$$

$$d_{2\theta} = \frac{\nabla_{\mathbf{u}} G(\mathbf{u}, \theta)^T \mathbf{u} - G(\mathbf{u}, \theta) + \beta_t \|\nabla_{\mathbf{u}} G(\mathbf{u}, \theta)\|}{\partial G(\mathbf{u}, \theta) / \partial \theta}. \quad (17)$$

This inverse-FORM algorithm based on the HLRF algorithm successfully provided inverse reliability solutions as demonstrated by the two examples in the work done by

Der Kiureghian et al. [11]. However, the algorithm converges slowly for highly nonlinear examples [18] and can be further improved by using the BFGS-based-hybrid algorithm that is proposed in the next section.

3. BFGS-Based Algorithm

3.1. Intermediate Algorithm. In this section, an intermediate algorithm that was developed by modifying the HLRF algorithm in inverse-FORM using the Broyden-Fletcher-Goldarb-Shanno (BFGS) update formula is explained. This intermediate algorithm uses calculation steps similar to those in inverse-FORM to solve an inverse reliability problem described in (2a), (2b), and (2c) incorporating the BFGS formula. The starting points for $\mathbf{u}$ and θ are adopted as $\mathbf{u}_0$ and θ_0 , respectively, and the values of $\mathbf{u}$ and θ are updated using the step direction vector, using (3). The step size λ_i in (3) is found such that the condition in (4) is satisfied.

The difference of this intermediate algorithm is as follows: in the calculation of the merit function $f(\mathbf{u}, \theta)$ in (5), $f_1(\mathbf{u}, \theta)$ is calculated using the following equation instead of (6), while $f_2(u, \theta)$ is calculated using (7) unchangingly:

$$f_1 = \frac{1}{2} \left\| \mathbf{B}_i^{-1} \mathbf{u} - \left[\frac{\nabla_{\mathbf{u}} G(\mathbf{u}, \theta)^T \mathbf{B}_i^{-1} \mathbf{u}}{\nabla_{\mathbf{u}} G(\mathbf{u}, \theta)^T \mathbf{B}_i^{-1} \nabla_{\mathbf{u}} G(\mathbf{u}, \theta)} \right] \mathbf{B}_i^{-1} \nabla_{\mathbf{u}} G(\mathbf{u}, \theta) \right\|^2 + \frac{1}{2} c_1 G(\mathbf{u}, \theta)^2, \quad (18)$$

where c_1 is calculated as follows:

$$c_1 = \frac{\max \left[\left\| \mathbf{B}_i^{-1} \mathbf{u} - \left[\nabla_{\mathbf{u}} G(\mathbf{u}, \theta)^T \mathbf{B}_i^{-1} \mathbf{u} / \nabla_{\mathbf{u}} G(\mathbf{u}, \theta)^T \mathbf{B}_i^{-1} \nabla_{\mathbf{u}} G(\mathbf{u}, \theta) \right] \mathbf{B}_i^{-1} \nabla_{\mathbf{u}} G(\mathbf{u}, \theta) \right\|^2, 0.001^2 \right]}{\max [G(\mathbf{u}, \theta)^2, 0.001^2]}. \quad (19)$$

Note that, in (18) and (19), the Hessian of the Lagrangian for the i th iteration is given by the BFGS matrix $\mathbf{B}_i$. The BFGS matrix in each iteration is updated using the BFGS update formula shown in (20). In the first step of the iterations however, the inverse of Hessian of the Lagrangian is taken as the identity matrix ($\mathbf{I}$) instead of using the following formula:

$$\mathbf{B}_{i+1}^{-1} = \mathbf{B}_i^{-1} + \left(1 + \frac{(\mathbf{q}^i)^T \mathbf{B}_i^{-1} \mathbf{q}^i}{(\mathbf{p}^i)^T \mathbf{q}^i} \right) \frac{\mathbf{p}^i (\mathbf{p}^i)^T}{(\mathbf{p}^i)^T \mathbf{q}^i} - \frac{\mathbf{p}^i (\mathbf{q}^i)^T \mathbf{B}_i^{-1} + \mathbf{B}_i^{-1} \mathbf{q}^i (\mathbf{p}^i)^T}{(\mathbf{p}^i)^T \mathbf{q}^i}, \quad (20)$$

where $\mathbf{p}^i$ and $\mathbf{q}^i$ are calculated using the following formula:

$$\begin{aligned} \mathbf{p}^i &= \mathbf{u}_{i+1} - \mathbf{u}_i, \\ \mathbf{q}^i &= \mathbf{u}_{i+1} - \mathbf{u}_i + (\nabla_{\mathbf{u}} G(\mathbf{u}_{i+1}, \theta_{i+1}) - \nabla_{\mathbf{u}} G(\mathbf{u}_i, \theta_i)) \varepsilon_i, \end{aligned} \quad (21)$$

where ε_i is calculated to be

$$\varepsilon_i = \frac{G(\mathbf{u}_i, \theta_i) - \nabla_{\mathbf{u}} G(\mathbf{u}_i, \theta_i)^T \mathbf{B}_i^{-1} \mathbf{u}_i}{\nabla_{\mathbf{u}} G(\mathbf{u}_i, \theta_i)^T \mathbf{B}_i^{-1} \nabla_{\mathbf{u}} G(\mathbf{u}_i, \theta_i)}. \quad (22)$$

The same as the inverse-FORM, the step direction vector ($\mathbf{d}$) in (3) is calculated using (9). In the calculation of $\mathbf{d}$, the values of a_1 , a_2 , $\mathbf{d}_1$, and $\mathbf{d}_2$ are calculated using (10), (11), (12), and (13), respectively.

Select target reliability index β_t .
 Select the starting point values for $\mathbf{u}$ and θ as $\mathbf{u}_0$ and θ_0 .
 $i = 0$; i is the iteration counter.
 $\mathbf{B}_i^{-1}$ is fixed as the identity matrix for the first iteration ($\mathbf{B}_0^{-1} = \mathbf{I}$).
 REPEAT (while the stopping criterion is not satisfied)
 (1) Compute f_1 and f_2 using Eqs. (18) and (7), respectively.
 (2) Compute a_1 and a_2 using Eqs. (10) and (11), respectively.
 (3) Compute $\mathbf{d}_1$ and $\mathbf{d}_2$ using Eqs. (12) and (13), respectively.
 (4) Compute the search direction vector ($\mathbf{d}$) using Eq. (9).
 (5) Compute the step size λ based on Eq. (4) by trying several values for λ .
 (6) Find $\mathbf{u}_{i+1}$ and θ_{i+1} using Eq. (3).
 (7) Update $\mathbf{B}_{i+1}^{-1}$ using the BFGS update formula as shown in Eq. (20).
 $i = i + 1$.
 END

ALGORITHM 1: Intermediate algorithm.

In the calculation of $\mathbf{d}_1$ using (12), $\mathbf{d}_{1u}$ is calculated using the following formula, instead of (14), while $d_{1\theta}$ is calculated unchangingly using (15):

$$\mathbf{d}_{1u} = \left[\frac{\nabla_{\mathbf{u}} G(\mathbf{u}, \theta)^T \mathbf{B}_i^{-1} \mathbf{u} - G(\mathbf{u}, \theta)}{\nabla_{\mathbf{u}} G(\mathbf{u}, \theta)^T \mathbf{B}_i^{-1} \nabla_{\mathbf{u}} G(\mathbf{u}, \theta)} \right] \mathbf{B}_i^{-1} \nabla_{\mathbf{u}} G(\mathbf{u}, \theta) - \mathbf{B}_i^{-1} \mathbf{u}. \quad (23)$$

In the calculation of $\mathbf{d}_2$ in (13), the values of $\mathbf{d}_{2u}$, and $d_{2\theta}$ are calculated using the following formulae instead of (16) and (17):

$$\begin{aligned} \mathbf{d}_{2u} &= -\beta_t \frac{\mathbf{B}_i^{-1} \nabla_{\mathbf{u}} G(\mathbf{u}, \theta)}{\|\mathbf{B}_i^{-1} \nabla_{\mathbf{u}} G(\mathbf{u}, \theta)\|} - \mathbf{B}_i^{-1} \mathbf{u}, \\ d_{2\theta} &= \frac{\nabla_{\mathbf{u}} G(\mathbf{u}, \theta)^T \mathbf{B}_i^{-1} \mathbf{u} - G(\mathbf{u}, \theta) + \beta_t \|\mathbf{B}_i^{-1} \nabla_{\mathbf{u}} G(\mathbf{u}, \theta)\|}{\partial G(\mathbf{u}, \theta) / \partial \theta}. \end{aligned} \quad (24)$$

A pseudocode for this intermediate algorithm is provided in Algorithm 1, in which it is noted that the BFGS update formula is used to update the inverse of Hessian in every iteration. The proposed use of the BFGS update formula increases the computational efficiency of this intermediate algorithm when compared to that of the inverse-FORM because it provides more optimized step changes.

The convergence criterion in the equation below is used to stop the iterative loop in Algorithm 1. It was originally proposed for inverse-FORM by Der Kiureghian et al. [11]:

$$\frac{\sqrt{\|\mathbf{u}_{i+1} - \mathbf{u}_i\|^2 + |\theta_{i+1} - \theta_i|^2}}{\sqrt{\|\mathbf{u}_{i+1}\|^2 + |\theta_{i+1}|^2}} \leq 0.001, \quad (25)$$

where $\mathbf{u}_i$ is the vector of design variables ($u_1, u_2, \dots, u_n$) in the i th iteration; $\|\mathbf{u}_i\|$ is its magnitude; $|\theta_i|$ is the absolute value of the unknown parameter in the i th iteration.

However, the intermediate algorithm is sometimes found to compromise the accuracy of the final solution as demonstrated through the numerical examples in Section 4. This is because the BFGS update formula does not always guarantee a rigorous convergence, and this is represented by fluctuations in the determinant value of the inverse of BFGS matrix through the numerical problems covered in this paper later. This fluctuation or the variation of the step size in each iteration affects the accuracy of the final solution apart from the improvement in the efficiency. To overcome this issue, the next section provides formulae for a BFGS-based hybrid algorithm, which has better accuracy and the stability in convergence by using both the original HLRF algorithm and the modified algorithm with BFGS update formula in a hybrid manner. This algorithm uses the same convergence criterion in (25).

3.2. BFGS-Based Hybrid Algorithm. The inverse-FORM algorithm and the intermediate algorithm mentioned in previous section become identical if the Hessian of the Lagrangian is approximated as the identity matrix in every iteration. By using the BFGS update formula in (20) to calculate the inverse of Hessian of the Lagrangian in the intermediate algorithm, the efficiency of the search is improved, but the final answer in this algorithm does not guarantee the same level of accuracy as that of inverse-FORM. Using the BFGS matrix results in larger step size and hence faster convergence. However, this convergence is less stable compared to the convergence generated by identity matrix, even though identity matrix results in slower convergence.

To show this in detail, the following numerical problem is considered, which is included as Problem 1a in the next section: the limit state function is given by $G_1(\mathbf{u}, \theta) = \exp(-\theta(\mathbf{u}_1 + 2\mathbf{u}_2 + 3\mathbf{u}_3)) - u_4 + 1.5$. The variables $\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3$, and $\mathbf{u}_4$ follow standard normal distribution and are uncorrelated. The starting point is taken to be 0.2 for all the design variables ($\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3, \mathbf{u}_4$), and 0.1 for the unknown parameter θ . The value of θ is calculated to be 0.3671 in the forward analysis, but we assume that it is unknown in the inverse reliability

```

Select target reliability index  $\beta_t$ .
Select the starting point values for  $\mathbf{u}$  and  $\theta$  as  $\mathbf{u}_0$  and  $\theta_0$ .
 $i = 0$ ;  $i$  is the iteration counter.
 $k = 0$ .
 $\mathbf{B}_i^{-1}$  is fixed as the identity matrix for the first iteration ( $\mathbf{B}_0^{-1} = I$ ).
REPEAT (while the convergence criterion is not satisfied)
  (1) The same as step (1) in the intermediate algorithm.
  (2) The same as step (2) in the intermediate algorithm.
  (3) The same as step (3) in the intermediate algorithm.
  (4) The same as step (4) in the intermediate algorithm.
  (5) The same as step (5) in the intermediate algorithm.
  (6) The same as step (6) in the intermediate algorithm.
  (7) If ( $k = 1$ ) OR determinant of inverse of BFGS matrix diverges away from 1.0
       $k = 1$ ;
      fix the inverse of Hessian as identity matrix ( $I$ ).
    else
      update inverse of Hessian as  $\mathbf{B}_{i+1}^{-1}$  using the BFGS update formula shown in Eq. (20).
      where  $\mathbf{B}_i^{-1}$  is the inverse of BFGS matrix in the  $i$ th iteration and  $k$  is the variable to
      keep track of the determinant of  $\mathbf{B}_i^{-1}$ .  $k$  has an initial value of 0.
    END
     $i = i + 1$ .
END

```

ALGORITHM 2: BFGS-based hybrid algorithm.

TABLE 1: Determinant of inverse of Hessian and limit state function value in each iteration, when the intermediate algorithm is used for Problem 1a.

Iteration no:	1	2	3	4	5	6
Determinant of inverse of Hessian	1.0	1.4227	1.7004	1.0497	1.0261	1.0009
$G_1(\mathbf{u}, \theta)$	2.1869	-0.0690	0.0256	0.0478	0.0138	0.0001

analysis. The reliability index (β) corresponding to this value of θ is 2.0, and it is taken to be the target reliability index (β_t) for the inverse reliability analysis.

The iterations for all the problems considered in this paper, including this problem, were stopped using the stopping criteria in (25). To trace the stability of the convergence of the updating process of the inverse of BFGS matrix, the determinant of the inverse of BFGS matrix and the limit state function values are monitored in Table 1. It shows that the determinant of inverse of Hessian is 1.0 in the first iteration as an initial value, and, for the second to sixth iterations, the value stably converges from 1.42274 to 1.0, except for the third iteration. Due to this exception in the third iteration, the value of the limit state function does not converge to 0 in the fourth iteration, and this step impedes the convergence of the overall analysis; this is different from inverse-FORM.

To avoid this, we propose further modifications resulting in a BFGS-based hybrid algorithm, in which the BFGS update formula is used only until the algorithm meets a determinant value that is neither 1.0 nor converging towards 1.0. For the subsequent iterations, the BFGS update formula is neglected, and the inverse of Hessian is fixed as unity. Note that this checking process always starts from the third iteration step, as the second iteration step cannot provide a clear indication on convergence or divergence. This technique is illustrated in Algorithm 2. As shown in Algorithm 2, a variable k is used

in the BFGS-based hybrid algorithm. This variable represents the condition for checking whether the inverse of Hessian of the Lagrangian being fixed to be the identity matrix or not and whether the BFGS update formula should be further used or not. The variable k 's initial value is zero and remains as zero until the condition for the inverse of Hessian matrix being fixed as the identity matrix is satisfied; then, the value of k is changed to one. For all subsequent iterations, the value of k is fixed as 1, and the inverse of Hessian is fixed as the identity matrix.

When we apply this hybrid algorithm to the above-mentioned numerical example, the changes in the determinant of the inverse of Hessian, the logic flow, and the action in each iteration are outlined in Table 2. The table shows that the inverse of Hessian is fixed to be the identity matrix from the third iteration according to the proposed process. By using inverse-FORM, the intermediate algorithm, and the BFGS-based hybrid algorithm, the results for θ are estimated as 0.3669, 0.3535, and 0.3668, respectively, while the true solution is 0.3671. It is seen that inverse-FORM and the BFGS-based hybrid algorithm show almost the same result but the intermediate algorithm shows a slightly different value. The results for β using these three methods are estimated as 2.0002, 2.0136, and 2.0003, respectively, while the true answer is 2.0000. Again, inverse-FORM and BFGS-based hybrid algorithm show almost the same result but

TABLE 2: Logic flow in each iteration of the BFGS-based hybrid algorithm for the considered example.

Iteration number	The determinant of the inverse of Hessian	Is $k = 1$?	Is $ \mathbf{B}_i^{-1} $ diverging away from 1.0?	Action
				The initial value of the inverse of Hessian is assumed to be the identity matrix The initial value of k is 0
1	1.0	No	No	The BFGS update formula is used to update the inverse of Hessian k remains unchanged
2	1.4227	No	Yes	The inverse of Hessian is fixed as the identity matrix The value of k is changed to 1
3	1.0	Yes	No need to check further	The inverse of Hessian is fixed as the identity matrix k remains unchanged
4	1.0	Yes	No need to check further	The inverse of Hessian is fixed as the identity matrix k remains unchanged
5	1.0	Yes	No need to check further	The inverse of Hessian is fixed as the identity matrix k remains unchanged
6	1.0	Yes	No need to check further	The inverse of Hessian is fixed as the identity matrix k remains unchanged
7	1.0	Yes	No need to check further	The inverse of Hessian is fixed as the identity matrix k remains unchanged

the intermediate algorithm shows slightly different value. A number of iterations required by inverse-FORM, the intermediate algorithm, and the BFGS-based hybrid algorithm are 8, 6, and 7, respectively. The computational efficiencies of the intermediate algorithm and the BFGS-based hybrid algorithm are better than that of inverse-FORM. This example illustrated the calculation steps and the comparison of the accuracies and computational efficiencies of the inverse reliability algorithms. Further comprehensive comparisons of the performance of these three methods are provided in the following section through various numerical examples.

4. Numerical Experiments

This section presents the comparisons of the efficiency and accuracies of the three algorithms for a total of 15 numerical problems. In each problem, a different limit state function represented by $G(\mathbf{u}, \theta)$ is considered, where $\mathbf{u}$ is the vector of design variables: $[u_1 \ u_2 \ u_3 \ \cdots \ u_n]^T$, and θ represents the unknown parameter. It should be noted that the individual design variables u_1, u_2, u_3 , and so forth are scalars, whereas $\mathbf{u}$ is a vector. The failure region is defined by the following equation:

$$G(\mathbf{u}, \theta) < 0. \quad (26)$$

In the inverse reliability analysis problems, a reliability level is given, which is represented by the target reliability index (β_t). The value of the unknown parameter (θ) needs to be

determined for the design point, at which $\|\mathbf{u}\| = \beta_t$ and $G(\mathbf{u}, \theta) = 0$.

The considered problems are selected or modified from the literature or newly constructed in this study. For the remaining of this section, first, the limit state function, the starting points for the design variables ($u_i, i = 1 \cdots n$), and the starting point for the unknown parameter (θ) are provided for these problems; second, comparisons of the efficiency and accuracy of the three algorithms for solving the problems are provided in Table 3; third, an extensive analysis of this comparison is presented by analyzing some of the problems.

Problem 1 (from Der Kiureghian et al. [11]). A limit state function is given as $G_1(\mathbf{u}, \theta) = \exp(-\theta(u_1 + 2u_2 + 3u_3)) - u_4 + 1.5$. The variables u_1, u_2, u_3 , and u_4 follow the standard normal distribution and are uncorrelated. Three different sets of starting points are considered for u_1, u_2, u_3, u_4 , and θ to check if the final results vary depending on the starting points. Three cases 1a, 1b, and 1c consider different starting points.

1a. The starting point is 0.2 for all design variables (u_1, u_2, u_3 , and u_4) and 0.1 for the unknown parameter θ . The true solution for the value of θ is set to be 0.3671. The reliability index (β) corresponding to this value of θ is 2.0 from forward analysis, and it is taken as the target reliability index (β_t) for the inverse reliability analysis.

TABLE 3: Comparison between inverse-FORM, the intermediate algorithm, and the BFGS-based hybrid algorithm.

Problem		True solution (from forward analysis)	Inverse-FORM	Intermediate algorithm	BFGS-based algorithm BFGS-based hybrid algorithm
1a	Number of iterations	—	8	6	7
	$G_1(\mathbf{u}, \theta)$	0	0	0	0
	θ	0.3671	0.3669	0.3535	0.3668
	β	2.0000	2.0002	2.0136	2.0003
1b	Number of iterations	—	53	4	42
	$G_1(\mathbf{u}, \theta)$	0	0	0.0059	0
	θ	0.3671	0.3666	0.3079	0.3666
	β	2.0000	2.0005	2.0538	2.0005
1c	Number of iterations	—	62	5	30
	$G_1(\mathbf{u}, \theta)$	0	0	0.006	0
	θ	0.3671	0.3666	0.3525	0.3666
	β	2.0000	2.0005	2.0082	2.0005
2a	Number of iterations	—	10	7	7
	$G_2(\mathbf{u}, \theta)$	0	0	0.0038	0.0002
	θ	0.4291	0.4282	0.38427	0.4287
	β	2.0000	2.0007	2.032	2.0003
2b	Number of iterations	—	53	4	42
	$G_2(\mathbf{u}, \theta)$	0	0	0.0054	0
	θ	0.4291	0.4284	0.3623	0.4285
	β	2.0000	2.0005	2.0519	2.0005
2c	Number of iterations	—	56	4	37
	$G_2(\mathbf{u}, \theta)$	0	0	0.0088	0
	θ	0.4291	0.4284	0.3834	0.4284
	β	2.0000	2.0005	2.0298	2.0005
3	Number of iterations	—	4	3	3
	$G_3(\mathbf{u}, \theta)$	0	0.4211	0.0730	0.0730
	θ	2.08	0.6986	2.0030	2.0030
	β	2.0000	2.0003	1.9875	1.9875
4	Number of iterations	—	3	4	3
	$G_4(\mathbf{u}, \theta)$	0	0.2535	0.0522	0.0550
	θ	15.2593	13.3816	14.8710	14.8457
	β	2.0000	1.9998	1.9962	1.9861
5	Number of iterations	—	2	2	2
	$G_5(\mathbf{x}, \theta)$	0	0	0	0
	θ	2.5	2.5	2.5	2.5
	β	2.5	2.5	2.5	2.5

TABLE 3: Continued.

Problem		True solution (from forward analysis)	Inverse-FORM	BFGS-based algorithm	
				Intermediate algorithm	BFGS-based hybrid algorithm
6a	Number of iterations	—	20	6	19
	$G_6(\mathbf{x}, \theta)$	0	0	0.0998	0
	θ	1.3838	1.3846	1.4905	1.3847
	β	2.0000	2.0007	2.0063	2.0008
6b	Number of iterations	—	8	4	7
	$G_6(\mathbf{x}, \theta)$	0	0	−0.0657	0
	θ	1.3838	1.3838	1.4252	1.3839
	β	1.7346	1.7346	1.8274	1.7347
7	Number of iterations	—	17	6	7
	$G_7(\mathbf{x}, \theta)$	0	0	0.0937	0
	θ	2.0000	2.0012	2.1017	2.0003
	β	1.5105	1.5115	1.5172	1.5107
8	Number of iterations	—	10	6	10
	$G_8(\mathbf{x}, \theta)$	0	0	0.0264	0
	θ	1.3992	1.4000	1.4258	1.3999
	β	2.0000	2.0007	2.0001	2.0007
9a	Number of iterations	—	61	7	57
	$G_9(\mathbf{x}, \theta)$	0	0	−0.0680	0
	θ	1.0	1.0007	1.0424	1.0007
	β	1.677	1.6776	1.7787	1.6776
9b	Number of iterations	—	9	5	9
	$G_9(\mathbf{x}, \theta)$	0	0	−0.0041	0
	θ	1.0	1.0002	1.0333	1.0001
	β	1.4671	1.4672	1.4998	1.4672
10	Number of iterations	—	16	4	16
	$G_{10}(\mathbf{x}, \theta)$	0	0	0.0137	0
	θ	6.0	5.9946	4.7864	5.9946
	β	2.0	1.9993	1.8610	1.9993
11	Number of iterations	—	78	4	69
	$G_{11}(\mathbf{x}, \theta)$	0	0	0.1934	0
	θ	2.3584	2.3570	2.2964	2.3571
	β	1.4	1.3995	1.2382	1.3995
12	Number of iterations	—	23	6	13
	$G_{12}(\mathbf{x}, \theta)$	0	0	0.0752	0
	θ	1.5551	1.5561	1.6531	1.5560
	β	1.0	1.0008	1.0174	1.0007

TABLE 3: Continued.

Problem		True solution (from forward analysis)	Inverse-FORM	BFGS-based algorithm	
				Intermediate algorithm	BFGS-based hybrid algorithm
13a	Number of iterations	—	34	6	33
	$G_{13}(\mathbf{x}, \theta)$	0	0	0.0057	0
	θ	0.3438	0.3443	0.3648	0.3443
	β	1.0	1.0004	1.0133	1.0004
13b	Number of iterations	—	12	5	11
	$G_{13}(\mathbf{x}, \theta)$	0	0	0.0161	0
	θ	0.3438	0.3440	0.4305	0.3440
	β	0.7901	0.7903	0.8477	0.7902
14	Number of iterations	—	14	4	14
	$G_{14}(\mathbf{x}, \theta)$	0	0	0.0928	0
	θ	1.922	1.9210	1.7578	1.9210
	β	1.0	0.9997	0.9072	0.9997
15	Number of iterations	—	119	4	108
	$G_{15}(\mathbf{x}, \theta)$	0	0	0.1391	0
	θ	1.6941	1.6933	1.6515	1.6933
	β	1.0	0.9997	0.8844	0.9997

1b. The starting point is 0.3 for all the design variables (u_1 , u_2 , u_3 , and u_4) and 0.3 for the unknown parameter θ . All the other conditions are the same.

1c. The starting point is 0.4 for all the design variables (u_1 , u_2 , u_3 , and u_4) and 0.4 for the unknown parameter θ . All the other conditions are the same.

Problem 2 (modified from Der Kiureghian et al. [11]). A limit state function is given as $G_2(\mathbf{u}, \theta) = \exp(-\theta(u_1 + 0.5u_2 + 3u_3)) - u_4 + 1.5$. The variables u_1 , u_2 , u_3 , and u_4 follow standard normal distribution and are uncorrelated. Three different starting points are considered for u_1 , u_2 , u_3 , u_4 , and θ to check if the final results vary according to different starting points. Three cases 2a, 2b, and 2c consider different starting points.

2a. The starting point is 0.2 for all the design variables (u_1 , u_2 , u_3 , and u_4) and 0.1 for the unknown parameter θ . The solution for θ is set to be 0.4291. The reliability index (β) corresponding to this value of θ is 2.0 in the forward analysis, and it is taken as the target reliability index (β_t) for the inverse reliability analysis.

2b. The starting point is 0.3 for all the design variables (u_1 , u_2 , u_3 , and u_4) and 0.3 for the unknown parameter θ . All the other conditions are the same.

2c. The starting point is 0.4 for all the design variables (u_1 , u_2 , u_3 , and u_4) and 0.4 for the unknown parameter θ . All the other conditions are the same.

Problem 3 (modified from Grooteman [25]). A limit state function is given as $G_3(\mathbf{u}, \theta) = 2 - \theta - 0.1u_1^2 + 0.06u_1^3$. The variable u_1 follows the standard normal distribution. The starting points considered for u_1 and θ are equally 0. The solution for θ is set to be 2.08. The reliability index (β) corresponding to this value of θ is 2.0 in the forward analysis, and it is taken as the target reliability index (β_t) for the inverse reliability analysis.

Problem 4 (modified from Grooteman [25]). A limit state function is given as $G_4(\mathbf{u}, \theta) = 2 + 0.015 \sum_{i=1}^9 (u_i^2 - \theta)$. The variables u_1 , u_2 , u_3 , u_4 , u_5 , u_6 , u_7 , u_8 , and u_9 follow the standard normal distribution and are uncorrelated. The starting points for u_i , $i = 1$ to 9, and θ are equally 0. The solution for θ is set to be 15.2593. The reliability index (β) corresponding to this value of θ is 2.0 in the forward analysis, and it is taken as the target reliability index (β_t) for the inverse reliability analysis.

Problem 5 (modified from Grooteman [25]). A limit state function is given as $G_5(\mathbf{x}, \theta) = \theta - 0.2357(x_1 - x_2) + 0.0046(x_1 + x_2 - 20)^4$. The variables x_1 and x_2 are normally distributed with mean values $\mu_1 = \mu_2 = 10$ and standard deviations $\sigma_1 = \sigma_2 = 3$. The variables are uncorrelated. The starting points considered for x_1 , x_2 , and θ are μ_1 , μ_2 , and 0, respectively. The solution for θ is set to be 2.5. The reliability index (β) corresponding to this value of θ is 2.5 in the forward analysis, and it is taken as the target reliability index (β_t) for the inverse reliability analysis.

Problem 6a. A limit state function is given as $G_6(\mathbf{x}, \theta) = 0.2x_1 + \exp(x_2 - 2) + \theta - 2$. The variables x_1 and x_2 are

normally distributed with mean values $\mu_1 = 10$ and $\mu_2 = 2$ and standard deviations $\sigma_1 = 5$ and $\sigma_2 = 1$. The variables are uncorrelated. The starting points considered for x_1 , x_2 , and θ are μ_1 , μ_2 , and 0, respectively. The solution for θ is set to be 1.3838. The reliability index (β) corresponding to this value of θ is 2.0 in the forward analysis, and it is taken as the target reliability index (β_t) for the inverse reliability analysis.

Problem 6b. The limit state function is the same as in Problem 6a. The variables, their mean values, standard deviations, and starting points are also the same. The variables are correlated with a correlation coefficient of 0.5 between them. The solution for θ is set to be 1.3838. The reliability index (β) corresponding to this value of θ is 2.0 in the forward analysis, and it is taken as the target reliability index (β_t) for the inverse reliability analysis.

Problem 7. A limit state function is given as $G_7(\mathbf{u}, \theta) = 0.2x_1 - 2 + \sin(0.2x_2 - 2) + \theta$. The variables x_1 and x_2 are normally distributed with mean values $\mu_1 = \mu_2 = 10$ and standard deviations $\sigma_1 = \sigma_2 = 5$. The variables are uncorrelated. The starting points considered for x_1 , x_2 , and θ are μ_1 , μ_2 , and 0, respectively. The solution for θ is set to be 2.0. The reliability index (β) corresponding to this value of θ is 1.5105 in the forward analysis, and it is taken as the target reliability index (β_t) for the inverse reliability analysis.

Problem 8. A limit state function is given as $G_8(\mathbf{x}, \theta) = 0.25x_1 + \sin(\exp(0.5x_2 - 2)) + \theta - 2.5$. The variables x_1 and x_2 are normally distributed with mean values $\mu_1 = 10$ and $\mu_2 = 4$ and standard deviations $\sigma_1 = 4$ and $\sigma_2 = 2$. The variables are uncorrelated. The starting point considered for x_1 , x_2 , and θ are μ_1 , μ_2 , and 0, respectively. The solution for θ is set to be 1.3992. The reliability index (β) corresponding to this value of θ is 2.0 in the forward analysis, and it is taken as the target reliability index (β_t) for the inverse reliability analysis.

Problem 9a. A limit state function is given as $G_9(\mathbf{x}, \theta) = 0.25x_1 - 2.5 + \exp(\sin(0.25x_2 - 2.5)) + \theta$. The variables x_1 and x_2 are normally distributed with mean values $\mu_1 = \mu_2 = 10$ and standard deviations $\sigma_1 = \sigma_2 = 4$. The variables are uncorrelated. The starting point considered for x_1 is μ_1 , for x_2 is μ_2 , and for θ is 0. The solution for θ is set to be 1.0. The reliability index (β) corresponding to this value of θ is 1.677 in the forward analysis, and it is taken as the target reliability index (β_t) for the inverse reliability analysis.

Problem 9b. The limit state function is the same as in Problem 9a. The variables, their mean values, standard deviations, and starting points are also the same. The variables are correlated with a correlation coefficient of 0.5 between them. The solution for θ is set to be 1.0. The reliability index (β) corresponding to this value of θ is 1.4671 in the forward analysis, and it is taken as the target reliability index (β_t) for the inverse reliability analysis.

Problem 10. A limit state function is given as $G_{10}(\mathbf{x}, \theta) = 0.5x_1 - 2 + \theta \sin(\exp(0.5x_2 - 2))$. The variables x_1 and x_2 are normally distributed with mean values $\mu_1 = \mu_2 = 4$ and standard deviations $\sigma_1 = \sigma_2 = 2$. The variables are

uncorrelated. The starting points considered for x_1 , x_2 , and θ are μ_1 , μ_2 , and 0, respectively. The solution for θ is set to be 6.0. The reliability index (β) corresponding to this value of θ is 2.0 in the forward analysis, and it is taken as the target reliability index (β_t) for the inverse reliability analysis.

Problem 11. A limit state function is given as $G_{11}(\mathbf{x}, \theta) = 0.25x_1 - 1.5 + \theta \exp(\sin(0.5x_2 - 2))$. The variables x_1 and x_2 are normally distributed with mean values $\mu_1 = 6$ and $\mu_2 = 4$ and standard deviations $\sigma_1 = 4$ and $\sigma_2 = 2$. The variables are uncorrelated. The starting points considered for x_1 , x_2 , and θ are μ_1 , μ_2 , and 0, respectively. The solution for θ is set to be 2.3584. The reliability index (β) corresponding to this value of θ is 1.4 in the forward analysis, and it is taken as the target reliability index (β_t) for the inverse reliability analysis.

Problem 12. A limit state function is given as $G_{12}(\mathbf{x}, \theta) = 0.2x_1 - 2 + \sin(0.2(x_2 + x_3) - 4) + \theta$. The variables x_1 , x_2 , and x_3 are normally distributed with mean values $\mu_1 = \mu_2 = \mu_3 = 10$ and standard deviations $\sigma_1 = \sigma_2 = \sigma_3 = 5$. The variables are uncorrelated. The starting points considered for x_1 , x_2 , x_3 , and θ are μ_1 , μ_2 , μ_3 , and 0, respectively. The solution for θ is set to be 1.5551. The reliability index (β) corresponding to this value of θ is 1.0 in the forward analysis, and it is taken as the target reliability index (β_t) for the inverse reliability analysis.

Problem 13a. A limit state function is given as $G_{13}(\mathbf{x}, \theta) = 0.25x_1 - 1.5 + \exp(\sin(0.2(x_2 + x_3) - 4)) + \theta$. The variables x_1 , x_2 , and x_3 are normally distributed with mean values $\mu_1 = 6$ and $\mu_2 = \mu_3 = 10$ and standard deviations $\sigma_1 = 4$ and $\sigma_2 = \sigma_3 = 5$. The variables are uncorrelated. The starting points considered for x_1 , x_2 , x_3 , and θ are μ_1 , μ_2 , μ_3 , and 0, respectively. The solution for θ is set to be 0.3438. The reliability index (β) corresponding to this value of θ is 1.0 in the forward analysis, and it is taken as the target reliability index (β_t) for the inverse reliability analysis.

Problem 13b. The limit state function is the same as in Problem 13a. The variables, their mean values, standard deviations, and starting points are also the same. The variables are correlated with a correlation coefficient of 0.5 between each pair of variables. The solution for θ is set to be 0.3438. The reliability index (β) corresponding to this value of θ is 0.7901 in the forward analysis, and it is taken as the target reliability index (β_t) for the inverse reliability analysis.

Problem 14. A limit state function is given as $G_{14}(\mathbf{x}, \theta) = 0.33x_1 - 3 + \theta \exp(0.2x_2 + 0.25x_3 - 4.5)$. The variables x_1 , x_2 , and x_3 are normally distributed with mean values $\mu_1 = 9$ and $\mu_2 = \mu_3 = 10$ and standard deviations $\sigma_1 = 3$, $\sigma_2 = 5$, and $\sigma_3 = 4$. The variables are uncorrelated. The starting points considered for x_1 , x_2 , x_3 , and θ are μ_1 , μ_2 , μ_3 , and 0, respectively. The solution for θ is set to be 1.922. The reliability index (β) corresponding to this value of θ is 1.0 in the forward analysis, and it is taken as the target reliability index (β_t) for the inverse reliability analysis.

Problem 15. A limit state function is given as $G_{15}(\mathbf{x}, \theta) = 0.25x_1 - 2 + \theta \exp(\sin(0.25x_2 + 0.5x_3 - 5.5))$. The variables x_1 , x_2 , and x_3 are normally distributed with mean values

$\mu_1 = 8$, $\mu_2 = 10$, and $\mu_3 = 6$ and standard deviations $\sigma_1 = 4$, $\sigma_2 = 4$, and $\sigma_3 = 2$. The variables are uncorrelated. The starting points considered for x_1 , x_2 , x_3 , and θ are μ_1 , μ_2 , μ_3 , and 0, respectively. The solution for θ is set to be 1.6941. The reliability index (β) corresponding to this value of θ is 1.0 in the forward analysis, and it is taken as the target reliability index (β_t) for the inverse reliability analysis.

Table 3 illustrates the comparison of the inverse reliability algorithms (inverse-FORM, the intermediate algorithm, and the BFGS-based hybrid algorithm) for the above problems. In this table, the following values are provided at the solution point:

- (i) limit state function $G(\mathbf{u}, \theta)$;
- (ii) the unknown parameter θ ;
- (iii) the reliability index β .

These values are presented along with the number of iterations required by the algorithm to arrive at the solution. The numerical values of $G(\mathbf{u}, \theta)$ and β presented under the column named *true solution* represent the values of the limit state function (zero) and reliability index at the design point, estimated from forward reliability analysis using first-order reliability methods (FORM). These values are what we are targeting at. We assume that the value of the parameter θ is known beforehand in the forward analysis. For this particular value of θ , the design point and its associated reliability index β are estimated.

The number of iterations taken to arrive at the design point is not presented for the forward analysis, as the purpose of presenting Table 3 is to compare the computational efficiencies and accuracies of the inverse reliability algorithms (inverse-FORM, the intermediate algorithm, and the BFGS-based hybrid algorithm) only. To compare their accuracies, the values of $G(\mathbf{u}, \theta)$ and θ are used as the benchmark values and presented under the column named *true solution*. The values of $G(\mathbf{u}, \theta)$ and θ obtained from each of the three inverse reliability algorithms are compared against the benchmark values. It is to be noted that, in Table 3, $G(\mathbf{u}, \theta)$ values less than 10^{-4} have been regarded as zero. The value of the target reliability index β obtained from the forward analysis is used as the target reliability index β_t for the all the three inverse reliability algorithms. The values of the target reliability index β at the final solution point are presented under respective columns of each of the three inverse reliability algorithms.

From the comparison of the three algorithms as shown in Table 3, the following is observed in general:

- (i) The intermediate algorithm is computationally more efficient than inverse-FORM as it takes comparatively a smaller number of iterations to solve an inverse reliability problem. The worse efficiency of inverse-FORM is due to the slower progress by using the smaller step size of its search direction vector in every iteration, compared to that of the search direction vector used by the intermediate algorithm. However, inverse-FORM is more accurate than the intermediate algorithm in estimating the solution point, for almost all the examples.
- (ii) It is not as much as to the intermediate algorithm but the BFGS-based hybrid algorithm is also more efficient than

FIGURE 1: Determinant of $\mathbf{B}_i^{-1}$ or $\mathbf{I}$ versus iteration number, compared to the three algorithms for Problem 7.

inverse-FORM. From Table 3, it is observed that the BFGS-based hybrid algorithm is considerably more efficient than inverse-FORM in solving highly nonlinear problems. However, the BFGS-based hybrid algorithm does not compromise on the accuracy of the final solution as the intermediate algorithm does. It can be observed from Table 3 that for most of the examples, the BFGS-based hybrid algorithm is either more accurate than or at least as accurate as inverse-FORM, while having better efficiency in general.

To provide further explanations about the advantages and limitations of the three algorithms, two of the problems considered in Table 3 (Problems 7 and 12) have been analyzed as follows.

Further Explanation about Problem 7. Figure 1 illustrates the identity inverse-Hessian matrix for inverse-FORM and the updating processes of the determinant of inverse of BFGS matrix ($\mathbf{B}_i^{-1}$) in the intermediate algorithm and the BFGS-based hybrid algorithm for Problem 7. The values for inverse-FORM are represented by dot-dash lines with circular markers; those for the intermediate algorithm are represented by solid lines with asterisk markers; and those for the BFGS-based hybrid algorithm are represented by dotted lines with square markers. When inverse-FORM is used, the determinant of the identity inverse-Hessian matrix is one for all iterations, but when the intermediate algorithm is used, the determinant of inverse of BFGS matrix is observed to fluctuate as shown in Figure 1. It starts from one in the first two iterations and increases to 1.2579 in the third iteration, and then to 1.2678 in the fourth iteration. It decreases to 1.0045 in the fifth iteration but again slightly increases to 1.0066 in the sixth iteration. This fluctuation is due to the updating formula used in the intermediate algorithm, and it affects the convergence of the result.

FIGURE 2: Unknown parameter versus iteration number for Problem 7, compared to the three algorithms.

Figure 2 shows the values of the unknown parameter (θ) for the three algorithms. In this figure, the true solution of θ that needs to be found is represented by dashed lines. It should be noted in this figure that the solution for θ obtained using the intermediate algorithm is less accurate when compared to that obtained using inverse-FORM. As shown in Figure 2, the final value of θ is 2.1017, when inverse-BFGS is used, although its true solution is 2.0. However, its final value is 2.0012 when inverse-FORM is used, which is much more accurate. This is because of the fluctuation of the determinant of inverse of BFGS matrix in the intermediate algorithm, due to which the algorithm sometimes arbitrarily meets the convergence criteria and suddenly stops the iteration even when the obtained solution is not close enough to the true solution.

To obtain a balance between accuracy and efficiency, the BFGS-based hybrid algorithm is introduced. As explained before, in this algorithm, when the determinant of inverse of BFGS matrix diverges away from unity, we fix the determinant to unity manually for the subsequent iterations. This process prevents further unnecessary fluctuation of the determinant. In Figure 1, the dotted line and square markers show that the determinant of inverse of BFGS matrix moves to 1.2579 in the third iteration. However, from the fourth iteration onwards, it is manually fixed to unity for all iterations. As a result of this fixation, the BFGS-based hybrid algorithm is still more efficient than inverse-FORM but it does not compromise the accuracy of the final solution, unlike the intermediate algorithm. As shown in Figure 2, it should be noted that the final result is obtained in just seven iterations using the BFGS-based hybrid algorithm, compared to the 17 iterations taken by inverse-FORM. At the same time, it should also be noted that the final θ value of 2.0003 obtained using the BFGS-based hybrid algorithm is marginally even

more accurate than the final value of 2.0012 obtained using inverse-FORM.

Further Explanation about Problem 12. Figure 3 illustrates the identity inverse-Hessian matrix for inverse-FORM and the updating processes of the determinant of inverse of BFGS matrix ($\mathbf{B}_i^{-1}$) based on the intermediate algorithm and the BFGS-based hybrid algorithm for Problem 12. Similar to the results for Problem 7, the determinant of inverse of BFGS matrix is observed to fluctuate. It increases from 1 to 1.4487 in the third iteration before it decreases to 1.0022 in the subsequent iterations, and then again it increases to 1.0517 in the sixth iteration. This fluctuation results in larger step size and faster convergence, and the intermediate algorithm takes just six iterations to arrive at the solution compared to the 23 taken by inverse-FORM. However, this fluctuation also decreases the accuracy of the final solution. As shown in Figure 2, the final value of θ is 1.6530, when the intermediate algorithm is used, although the true solution is 1.5551. However, its final value is 1.5561 when inverse-FORM is used, which is more accurate.

Figure 4 shows the corresponding search values for the unknown parameter (θ), in the respective iterations. The BFGS-based hybrid algorithm has an optimum balance between efficiency and accuracy. In this algorithm, once the determinant of inverse of BFGS matrix diverges away from unity, it is fixed manually to unity for the subsequent iterations. In Figure 3, the dotted line with square markers shows that the determinant of inverse of BFGS matrix becomes 1.4487 in the third iteration. From the fourth iteration onwards, it starts converging towards the unity (the identity matrix $\mathbf{I}$) as it reaches 1.0222 in the fifth iteration. However, in the sixth iteration, it increases to 1.0517 and starts to diverge from the identity matrix. Hence, it is manually fixed to unity for all the other subsequent iterations. As a result of this fixation, the BFGS-based hybrid algorithm is still more efficient than inverse-FORM but does not compromise the accuracy of the final solution unlike the intermediate algorithm. As shown in Figure 4, the final result is obtained in just 13 iterations using the BFGS-based hybrid algorithm, compared to the 17 iterations taken by inverse-FORM. Moreover, as shown by in Figure 4, the final θ value is 1.5560 when the BFGS-based hybrid algorithm is used. This value is marginally more accurate than the final value of 1.5561 obtained using inverse-FORM.

It should also be informed that there are cases where all the three algorithms fail to get a solution due to a divergence issue. When the algorithms were applied to few nonlinear problems, they were found to converge to a point that is not on the limit state boundary. In other words, the solution point does not satisfy the condition given by (2c) and hence cannot be deemed as the design point. We observed that all three algorithms failed in the following example problem.

Problem 16. A limit state function is given as $G_{16}(\mathbf{x}, \theta) = x_1 x_2 - \theta$. The variables x_1 and x_2 are lognormally distributed with mean values $\mu_1 = 38$ and $\mu_2 = 54$ and standard deviations $\sigma_1 = 3.8$ and $\sigma_2 = 2.7$. The variables are uncorrelated. The starting point considered for x_1 is μ_1 , for

FIGURE 3: Determinant of $\mathbf{B}_i^{-1}$ or $\mathbf{I}$ versus iteration number, compared to the three algorithms for Problem 12.

FIGURE 4: Unknown parameter versus iteration number for Problem 12, compared to the three algorithms.

x_2 is μ_2 , and for θ is 0. The solution for θ is set to be 1140. The reliability index (β) corresponding to this value of θ is 5.2126 in the forward analysis, and it is taken as the target reliability index (β_t) for the inverse reliability analysis.

Inverse-FORM and the intermediate algorithm required just four and two iterations, respectively, to arrive at their final points. However, these final points do not represent the true solution. Figures 5 and 6 show that the final value of θ obtained using either algorithm is far away from its true

solution 1140. In addition, the value of the limit state function ($G(\mathbf{u}, \theta)$) does not converge to zero before the convergence criteria are met by the intermediate algorithm, as shown in Table 4. Moreover, as only two iterations were required to arrive at the solution, the inverse of the Hessian could not be analyzed. Since inverse-FORM and the intermediate algorithm failed to solve the problem, an optimum balance between the two could not be established and the BFGS-based hybrid algorithm cannot improve the intermediate algorithm

TABLE 4: Determinant of inverse of Hessian and limit state function value in each iteration, when the intermediate algorithm is used for Problem 16.

Iteration number	1	2
Determinant of inverse of Hessian	1.0	1.5858
$G_1(\mathbf{u}, \theta)$	2052	113.7

FIGURE 5: Limit state function and unknown parameter values for Problem 16 when inverse-FORM is used.

FIGURE 6: Limit state function and unknown parameter values for Problem 16 when the intermediate algorithm is used.

in such problems. It is important to note that this problem could not be solved by inverse-FORM either. Hence, it is concluded that the inherent limitation of inverse-FORM in solving certain highly nonlinear limit states is reflected in the limitation to solve this problem by the other two proposed methods.

5. Conclusions

An inverse reliability analysis algorithm based on inverse-FORM, using the BFGS update formula is proposed. The performance of this algorithm is compared with that of

inverse-FORM through 15 numerical experiments. From the analysis of the numerical experiments, the following is concluded:

- (1) The Hessian of the Lagrangian in inverse-FORM is approximated as the identity matrix $\mathbf{I}$ for each iteration. This makes the step size considerably small for all iterations. In other words, the number of iterations needed to attain the final solution increases, and it decreases the computational efficiency.
- (2) On the other hand, the intermediate algorithm has better efficiency than inverse-FORM by using the BFGS update formula in each iteration. The step size is relatively large for all iterations. However, this algorithm sometimes compromises the accuracy of the final solution, which is shown by the fluctuating values of inverse of Hessian of the Lagrangian near the design point.
- (3) For an optimum balance between efficiency and accuracy, the BFGS-based hybrid algorithm is proposed to use. In this algorithm, the inverse of BFGS matrix is used as the inverse of Hessian of the Lagrangian as long as it keeps moving towards unity ($\mathbf{I}$). However, once it starts to diverge from unity, it is fixed to unity. Hence, few iterations use $\mathbf{I}$ and few iterations use the BFGS update formula. This algorithm has similar or better accuracy, compared to inverse-FORM, for the example problems considered in Section 4. Moreover, it is more efficient compared to inverse-FORM.
- (4) For certain highly nonlinear problems, the inverse-FORM algorithm may not achieve convergence. This limitation still applies to the intermediate algorithm and the BFGS-based hybrid algorithm.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors wish to acknowledge the help of Gislaïne Perićaro at University of the State of Paraná, Brazil, for sending the MATLAB code of the HLRF-BFGS algorithm. This research was supported by the National Natural Science Foundation of China (Grant no. 51650110509).

References

- [1] D. I. Blockley, *Engineering Safety*, McGraw-Hill, London, England, 1992.
- [2] O. Ditlevsen and H. O. Madsen, *Structural Reliability Methods*, Wiley, New York, NY, USA, 1996.
- [3] M. G. Stewart and R. E. Melchers, *Probabilistic Risk Assessment of Engineering System*, Chapman Hall, London, England, 1997.
- [4] R. E. Melchers, *Structural Reliability Analysis and Prediction*, John Wiley Sons, Inc, Hoboken, NJ, United States, 1999.
- [5] R. Mínguez, E. Castillo, and A. S. Hadi, "Solving the inverse reliability problem using decomposition techniques," *Structural Safety*, vol. 27, no. 1, pp. 1–23, 2005.

- [6] H. O. Madsen, S. Krenk, and N. C. Lind, *Methods of Structural Safety*, Courier Corporation, Chelmsford, Mass, USA, 2006.
- [7] BSI (British Standards Institution). BS EN 1992:2002: Eurocode: Basis of structural design. BSI, London, UK, 2002.
- [8] H. Li and R. O. Foschi, "An inverse reliability method and its application," *Structural Safety*, vol. 20, no. 3, pp. 257–270, 1998.
- [9] P. Ramu, X. Qu, B. D. Youn, R. T. Haftka, and K. K. Choi, "Inverse reliability measures and reliability-based design optimisation," *International Journal of Reliability and Safety*, vol. 1, no. 1-2, pp. 187–205, 2006.
- [10] S. R. Winterstein, T. C. Ude, C. A. Cornell, P. Bjerager, and S. Haver, "Environmental parameters for extreme response: inverse FORM with omission factors," in *Proceedings of 6th International Conference on Structural Safety and Reliability*, Innsbruck, Austria, 1993.
- [11] A. Der Kiureghian, Y. Zhang, and C.-C. Li, "Inverse reliability problem," *Journal of Engineering Mechanics*, vol. 120, no. 5, pp. 1154–1159, 1994.
- [12] A. Hasofer, "Exact second-moment reliability for sum of products failure equation," *Journal of Structural Mechanics*, vol. 3, no. 1, pp. 97–106, 1974.
- [13] R. Rackwitz and B. Flessler, "Structural reliability under combined random load sequences," *Computers and Structures*, vol. 9, no. 5, pp. 489–494, 1978.
- [14] P.-L. Liu and A. der Kiureghian, "Optimization algorithms for structural reliability," *Structural Safety*, vol. 9, no. 3, pp. 161–177, 1991.
- [15] L. M. Fitzwater, C. A. Cornell, and P. S. Veers, "Using environmental contours to predict extreme events on wind turbines," in *Proceedings of the ASME 2003 Wind Energy Symposium*, pp. 244–258, Reno, Nev, USA.
- [16] K. Saranyasoontorn and L. Manuel, "Efficient models for wind turbine extreme loads using inverse reliability," *Journal of Wind Engineering and Industrial Aerodynamics*, vol. 92, no. 10, pp. 789–804, 2004.
- [17] S. Saha and C. S. Manohar, "Inverse reliability based structural design for system dependent critical earthquake loads," *Probabilistic Engineering Mechanics*, vol. 20, no. 1, pp. 19–31, 2005.
- [18] G. A. Perićaro, S. R. Santos, A. A. Ribeiro, and L. C. Matioli, "HLRF-BFGS optimization algorithm for structural reliability," *Applied Mathematical Modelling*, vol. 39, no. 7, pp. 2025–2035, 2015.
- [19] M. J. D. Powell, "VMCWD: A FORTRAN Subroutine for Constrained Optimization. Report DAMTP 1982/NA4, Cambridge University, United Kingdom".
- [20] K. Schittkowski, "On the convergence of a sequential quadratic programming method with an augmented Lagrangian line search function," *Mathematische Operationsforschung und Statistik Series Optimization*, vol. 14, no. 2, pp. 197–216, 1983.
- [21] D. G. Luenberger and Y. Ye, *Linear and Nonlinear Programming*, vol. 228, Springer, Berlin, Germany, 2016.
- [22] R. Rackwitz, "Reliability analysis—a review and some perspective," *Structural Safety*, vol. 23, no. 4, pp. 365–395, 2001.
- [23] C. Zhu, R. H. Byrd, P. Lu, and J. Nocedal, "Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization," *Association for Computing Machinery. Transactions on Mathematical Software*, vol. 23, no. 4, pp. 550–560, 1997.
- [24] J. Ahn and J. H. Kwon, "An efficient strategy for reliability-based multidisciplinary design optimization using BLISS," *Structural and Multidisciplinary Optimization*, vol. 31, no. 5, pp. 363–372, 2006.
- [25] F. Grooteman, "Adaptive radial-based importance sampling method for structural reliability," *Structural Safety*, vol. 30, no. 6, pp. 533–542, 2008.