Struct Multidisc Optim (2008) 35:245–261
DOI 10.1007/s00158-007-0133-3

RESEARCH PAPER

Design sensitivity and reliability-based structural optimization by univariate decomposition

S. Rahman · D. Wei

Received: 26 October 2006 / Revised: 10 March 2007 / Accepted: 19 March 2007 / Published online: 17 May 2007

Abstract This paper presents a new univariate decomposition method for design sensitivity analysis and reliability-based design optimization of mechanical systems subject to uncertain performance functions in constraints. The method involves a novel univariate approximation of a general multivariate function in the rotated Gaussian space for reliability analysis, analytical sensitivity of failure probability with respect to design variables, and standard gradient-based optimization algorithms. In both reliability and sensitivity analyses, the proposed effort has been reduced to performing multiple one-dimensional integrations. The evaluation of these one-dimensional integrations requires calculating only conditional responses at selected deterministic input determined by sample points and Gauss–Hermite integration points. Numerical results indicate that the proposed method provides accurate and computationally efficient estimates of the sensitivity of failure probability, which leads to accurate design optimization of uncertain mechanical systems.

Keywords Reliability · Design sensitivity · Univariate decomposition · Most probable point · Stochastic optimization

S. Rahman (✉)
Department of Mechanical and Industrial Engineering,
The University of Iowa,
Iowa City, IA 52242, USA
e-mail: rahman@engineering.uiowa.edu
url: http://www.engineering.uiowa.edu/~rahman

D. Wei
Caterpillar Technical Center,
Mossville, IL 61552, USA

1 Introduction

Reliability and optimization are two key elements of engineering design of structural and mechanical systems. Reliability-based design optimization (RBDO) is a mathematical framework for solving optimization problems in the presence of uncertainty, typically manifested by probabilistic description embedded in objective and/or constraint functions. With new models and formulations appearing almost every year, RBDO has, indeed, evolved as a viable paradigm for engineering design under uncertainty.

Existing RBDO processes involving constraints with random description can be broadly classified based on the method employed in performing the underlying reliability analysis: (1) the first-order reliability method (FORM; Enevoldsen and Sorensen 1994; Kuschel and Rackwitz 1997; Kirjner-Neto et al. 1998; Chiralaksanakul and Mahadevan 2005; Tu et al. 1999; Agarwal and Renaud 2006) and (2) simulation methods (Royset and Polak 2004; Qu and Haftka 2004). The FORM-based approach involves linearization of the performance function associated with each constraint at a most probable point (MPP), which is located by solving a deterministic constrained optimization problem. A traditional FORM-based optimization requires nested design and reliability iteration loops and, therefore, constitutes a double loop (Enevoldsen and Sorensen 1994). For the double-loop RBDO, two equivalent approaches, known as the reliability index and the performance measure approaches, have emerged (Tu et al. 1999). In fact, most double-loop methods available today belong to one of these two approaches. Nevertheless, a double-loop method is expensive because, for each design (outer) iteration, a set of reliability (inner) iterations involving costly function evaluations has to be generated for locating MPP. To

overcome high computational expense, single-loop formulations exploiting the Karush–Kuhn–Tucker optimality condition at MPP have appeared (Kuschel and Rackwitz 1997; Liang et al. 2004). These single-loop methods, which satisfy constraints only at an optimal point, significantly improve the efficiency of an RBDO process. To avoid nested loops, a sequential optimization method has also been introduced (Du and Chen 2004). However, a fundamental requirement of existing double- and single-loop methods is FORM, which may provide inaccurate reliability estimates for highly nonlinear performance functions. In fact, FORM and its quadratic counterpart, the second-order reliability method (SORM), have two major sources of errors. First, FORM/SORM provide strictly asymptotic solutions that are valid and useful when the distance between the origin and MPP approaches infinity. For non-asymptotic applications involving a highly nonlinear performance function, its linear or quadratic approximation may not be adequate; therefore, resultant FORM/SORM predictions should be interpreted with caution (Bjerager 1988; Nie and Ellingwood 2000). Second, if multiple MPPs exist in either asymptotic or non-asymptotic applications, classical FORM/SORM may yield erroneous failure probability estimates (Der Kiureghian and Dakessian 1998). Therefore, methods that can account for one or both sources of these potential errors are required. Indeed, recent results from Zou and Mahadevan (2006) and the present work show that FORM-based RBDO processes may produce infeasible or inaccurate designs. In contrast, simulation-based RBDO approaches, such as those employing Monte Carlo simulation, do not exhibit such limitations and, therefore, can be utilized for highly accurate reliability analysis. However, simulation-based RBDO methods generally require considerably more extensive calculations than the former methods and are consequently limited to solving simple mathematical problems.

Recently, the authors have developed new decomposition methods for solving highly nonlinear reliability problems (Xu and Rahman 2005; Rahman and Wei 2006; Wei and Rahman 2007). A major advantage of these decomposition methods, based on the mean point (Xu and Rahman 2005) or MPP (Rahman and Wei 2006; Wei and Rahman 2007) of random input as reference points, over FORM/SORM is that higher-order approximations of performance functions can be achieved using function values alone. Employing the mean point as the reference point in the decomposition methods has the advantage that a performance function can be approximated without requiring the MPP, which can sometimes be difficult to locate for a certain class of reliability problems. However, the mean-point-based decomposition methods may also require bivariate or higher-variate approximations to adequately represent a performance function, leading to quadratic or higher-order cost scaling with respect to the number of random variables. An attractive alternative is the MPP-based univariate decomposition method that maintains a linear cost scaling, however, with the additional expense of identifying the MPP. In fact, the MPP-based univariate method developed in authors' previous work was demonstrated to yield both accurate and computationally efficient solutions of reliability. The present work extends previous work in solving RBDO problems.

This paper presents a univariate decomposition method for the RBDO of mechanical systems. The method involves (1) higher-order, univariate approximation of performance functions for reliability analysis; (2) analytical sensitivities of failure probability with respect to design variables; and (3) standard gradient-based optimization algorithms. Section 2 gives a brief exposition of the RBDO formulation studied. Section 3 summarizes the MPP-based univariate decomposition method, develops new sensitivity equations for design variables, and presents a design optimization algorithm with computational flow. Two sets of examples, each involving mathematical functions and structural/solid-mechanics problems, illustrate the sensitivity analysis and RBDO method developed in Section 4. Comparisons have been made with alternative FORM/SORM and simulation-based methods to evaluate the accuracy and computational efficiency of the new RBDO method. Finally, Section 5 provides conclusions and future outlook.

2 Reliability-based design optimization

2.1 General RBDO problem

The mathematical formulation of a general stochastic optimization problem $\mathcal{P}_1$ involving a single objective function, and $1 \le K < \infty$ constraint functions entail the statement

$$\mathcal{P}_1 : \begin{cases} \min\limits_{\boldsymbol{d}\in\mathcal{D}\subseteq\mathbb{R}^{\mathcal{M}}} & c_0(\boldsymbol{d}) \equiv \mathbb{B}[f_0(\boldsymbol{X};\boldsymbol{d})] \\ \text{subject to} & c_k(\boldsymbol{d}) \equiv P[\boldsymbol{X}\in\Omega_{F,k}(\boldsymbol{d})] \le p_k; k = 1,\cdots,K \\ & d_{iL} \le d_i \le d_{iU}; i = 1,\cdots,M \end{cases} \tag{1}$$

in which $\boldsymbol{d} = \{d_1,\cdots,d_M\}^T \in \mathcal{D}$ is an M-dimensional design vector with a non-empty, closed set $\mathcal{D} \subseteq \mathbb{R}^{\mathcal{M}}$; $\boldsymbol{X} = \{X_1,\cdots,X_N\}^T \in \mathbb{R}^N$ is an N-dimensional random vector

with the joint probability density function $f_x(\boldsymbol{x})$ defined on a probability space $(\Omega, \mathcal{F}, P)$, where Ω is the sample space, $\mathcal{F}$ is the σ-algebra, and P is the probability measure; $\Omega_{F,k}(\boldsymbol{d}) \subseteq \Omega, k = 1, \cdots, K$ is the kth failure domain that may depend on $\boldsymbol{d}$, and $0 \le p_k \le 1, k = 1, \cdots, K$ are target failure probabilities, and d_{iL} and d_{iU} are respectively the lower and upper bounds of d_i. The design vector $\boldsymbol{d}$ can be deterministic parameters of objective and constraint functions and/or distribution parameters of $\boldsymbol{X}$ (e.g., mean of $\boldsymbol{X}$). The objective function c_0 is obtained by applying an appropriate risk functional $\mathbb{B} : f_{x \to \mathbb{R}}$ on a random state function $f_0(\boldsymbol{X}; \boldsymbol{d})$. For example, a common characterization of c_0 obtained by applying the expectation operator $\mathbb{E} : f_X \to \mathbb{R}$ is $c_0(\boldsymbol{d}) = \mathbb{E}[f_0(\boldsymbol{X}; \boldsymbol{d})]$, which involves statistical moment analysis. In contrast, the constraint function c_k, depicted in (1), requires reliability analysis. For component-reliability analysis, the failure domain $\Omega_{F,k} = \{\boldsymbol{x} : g_k(\boldsymbol{x}; \boldsymbol{d}) \le 0\}$, where $g_k(\boldsymbol{x}; \boldsymbol{d})$ is a single performance function for each constraint. Similar performance functions can be defined for a system-reliability analysis. Equation (1) defines a generic, single-objective RBDO problem.

2.2 Special RBDO problem

In engineering applications, RBDO is commonly formulated assuming a deterministic state function as the objective function and calculating component failure probabilities in constraint functions, leading to problem $\mathcal{P}_2$ with the mathematical statement

$$\mathcal{P}_2 : \begin{cases} \min\limits_{\boldsymbol{d} \in \mathcal{D}} & c_0(\boldsymbol{d}) \\ \text{subject to} & c_k(\boldsymbol{d}) \equiv \mathcal{P}[g_k(\boldsymbol{X}; \boldsymbol{d}) < 0] \le p_i; k = 1, \cdots, K \\ & d_{iL} \le d_i \le d_{iU}; i = 1, \cdots, M \end{cases} \tag{2}$$

which is a special case of problem $\mathcal{P}_1$. Solving problem $\mathcal{P}_2$ requires only component reliability analysis in evaluating constraints and is the focus of the current paper. The scope of problem $\mathcal{P}_2$ can be expanded by including constraints involving system-reliability analysis, which is not considered in the present study. The optimal solution is denoted by $\boldsymbol{d}^* \in \mathbb{R}^M$.

3 Univariate decomposition method

Consider a continuous, differentiable, real-valued performance function $g_k(\boldsymbol{x}; \boldsymbol{d}) = 0$ that depends on $\boldsymbol{x} = \{x_1, \cdots, x_N\}^T \in \mathbb{R}^N$ and $\boldsymbol{d} = \{d_1, \cdots, d_M\}^T \in \mathbb{R}^M$. If $\boldsymbol{u} = \{u_1, \cdots, u_N\}^T \in \mathbb{R}^N$ is the standard Gaussian space, let u_k^* denote the MPP or beta point, which is the closest point on the limit-state surface to the origin. The MPP has a distance $\beta_k(\boldsymbol{d})$, which is commonly referred to as the Hasofer–Lind reliability index (Madsen et al. 1986; Rackwitz 2001), is determined by a standard, nonlinear, constrained optimization. Construct an orthogonal matrix $\boldsymbol{R}_k \in \mathbb{R}^{N \times N}$ whose Nth column is $\boldsymbol{\alpha}_k^* \equiv \boldsymbol{u}_k^* / \beta_k$, i.e., $\boldsymbol{R}_k = \left[\boldsymbol{R}_{k,1} \middle| \boldsymbol{\alpha}_k^*\right]$, where $\boldsymbol{R}_{k,1} \in \mathbb{R}^{N \times N-1}$ satisfies $\boldsymbol{\alpha}_k^{*T} \boldsymbol{R}_{k,1} = \boldsymbol{0} \in \mathbb{R}^{1 \times N-1}$. The matrix $\boldsymbol{R}_{k,1}$ can be obtained, for example, by the Gram–Schmidt orthogonalization. For an orthogonal transformation $\boldsymbol{u} = \boldsymbol{R}_k \boldsymbol{v}_k$, let $\boldsymbol{v}_k = \{v_{k,1}, \cdots, v_{k,N}\}^T \in \mathbb{R}^N$ represent the rotated Gaussian space with the associated MPP $\boldsymbol{v}_k^* = \{v_{k,1}^*, \cdots, v_{k,N-1}^*, v_{k,N}^*\}^T = \{0, \cdots, 0, \beta_k\}^T$. The transformed limit states $h_k(\boldsymbol{u}; \boldsymbol{d}) = 0$ or $y_k(\boldsymbol{v}_k; \boldsymbol{d}) = 0$ are, therefore, the maps of the original performance function $g_k(\boldsymbol{x}; \boldsymbol{d}) = 0$ in the standard Gaussian space ($\boldsymbol{u}$ space) and the rotated Gaussian space ($\boldsymbol{v}_k$ space), respectively, as shown in Fig. 1 for N=2.

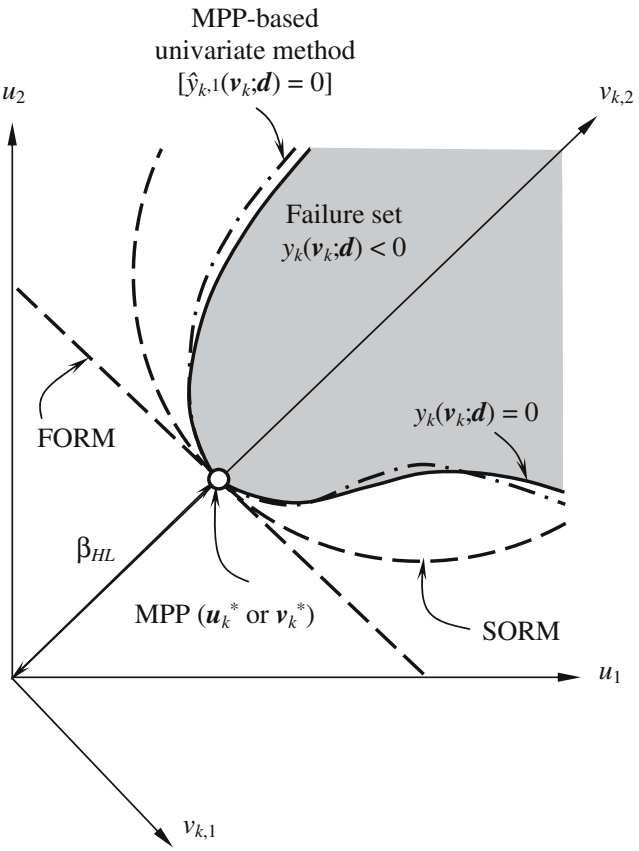

Fig. 1 Various approximations of the performance function of the kth constraint

3.1 Reliability analysis

3.1.1 MPP-based univariate decomposition of performance function

Consider a univariate approximation of $y_k(\boldsymbol{v}_k;\boldsymbol{d})$, denoted by (Rahman and Wei 2006)

$$\widehat{y}_{k,1}(\boldsymbol{v}_k;\boldsymbol{d}) \equiv \widehat{y}_{k,1}\left(v_{k,1},\cdots,v_{k,N};\boldsymbol{d}\right) = \sum_{i=1}^{N} y_{k,i}\left(v_{k,i};\boldsymbol{d}\right) - (N-1)y_k(\boldsymbol{v}_k^*;\boldsymbol{d}) \tag{3}$$

where $y_k\left(\boldsymbol{v}_k^*;\boldsymbol{d}\right) \equiv y_k(0,\cdots,0,\beta_k;\boldsymbol{d})$ and $y_{k,i}\left(v_{k,i};d\right) \equiv y_k\left(0,\cdots,0,v_{k,i},0,\cdots,0,\beta_k;\boldsymbol{d}\right); i=1,N$ are univariate component functions. From the authors' past work (Xu and Rahman 2005), it can be shown that the univariate approximation $\widehat{y}_{k,1}(\boldsymbol{v}_k;\boldsymbol{d})$ leads to a residual error $y_k(\boldsymbol{v}_k;\boldsymbol{d})-\widehat{y}_{k,1}(\boldsymbol{v}_k;\boldsymbol{d})$, which includes contributions from terms of dimension two and higher. For a sufficiently smooth $y_k(\boldsymbol{v}_k;\boldsymbol{d})$ with a convergent Taylor series, the coefficients associated with higher-dimensional terms are usually much smaller than those associated with one-dimensional terms. As such, higher-dimensional terms contribute less to the function and, therefore, can be neglected. Nevertheless, (3) includes all higher-order *univariate* terms. In contrast, FORM entails a univariate approximation retaining only *linear* terms. Hence, (3) should provide in general a higher-order approximation of the performance function than FORM. The curvature-fitted SORM has cross-terms, but is limited to a *quadratic* approximation. If the second-order cross-terms are negligibly small, the univariate decomposition may also provide a better approximation than SORM.

3.1.2 Failure probability analysis for constraint evaluations

Because $y_k\left(\boldsymbol{v}_k^*;\boldsymbol{d}\right)=0$, the univariate approximation in (3) can be rewritten as

$$\widehat{y}_{k,1}(\boldsymbol{v}_k;\boldsymbol{d}) = y_{k,N}\left(v_{k,N};\boldsymbol{d}\right) + \sum_{i=1}^{N-1} y_{k,i}\left(v_{k,i};\boldsymbol{d}\right) \tag{4}$$

where, due to the rotational transformation of the coordinates (see Fig. 1), the Nth univariate component function $y_{k,N}\left(v_{k,N};\boldsymbol{d}\right)$ is expected to be a linear or a weakly nonlinear function of v_N. In fact, $y_{k,N}\left(v_{k,N};\boldsymbol{d}\right)$ is linear with respect to $v_{k,N}$ in classical FORM/SORM approximations of a performance function in the $\boldsymbol{v}_k$ space. Hence, consider a linear approximation $y_{k,N}\left(v_{k,N};\boldsymbol{d}\right) = \beta_k(\boldsymbol{d}) + b_{k,0}(\boldsymbol{d}) + b_{k,1}(\boldsymbol{d})v_{k,N}$ and a quadratic approximation $y_{k,N}\left(v_{k,N};\boldsymbol{d}\right) = \beta_k(\boldsymbol{d}) + b_{k,0}(\boldsymbol{d}) + b_{k,1}(\boldsymbol{d})v_{k,N} + b_{k,2}(\boldsymbol{d})v_{k,N}^2$, respectively, where coefficients $b_{k,m}\in\mathbb{R}; m=0,1,2$ are obtained by least-squares approximations from exact or numerically simulated responses $\left\{y_{k,N}\left(v_{k,N}^{(1)};\boldsymbol{d}\right),\cdots,y_{k,N}\left(v_{k,N}^{(n)};\boldsymbol{d}\right)\right\}$ at n sample points along the $v_{k,N}$ coordinate. The function $y_{k,N}\left(v_{k,N};\boldsymbol{d}\right)$ also depends on $\beta_k(\boldsymbol{d})$, which is included in both approximations. Invoking these linear and quadratic approximations of $y_{k,N}\left(v_{k,N};\boldsymbol{d}\right)$ and noting that $V_{k,N}$ follows standard Gaussian distribution, the component failure probability embedded in the kth constraint can be expressed by (Wei and Rahman 2007)[1]

$$c_k(\boldsymbol{d}) = P[y_k(\boldsymbol{V}_k;\boldsymbol{d})<0] \cong P\left[\widehat{y}_{k,1}(\boldsymbol{V}_k;\boldsymbol{d})<0\right] \cong \begin{cases} \mathbb{E}\left[\Phi\left(\dfrac{-\beta_k(\boldsymbol{d})-b_{k,0}(\boldsymbol{d})-\sum_{i=1}^{N-1}y_{k,i}\left(V_{k,i};\boldsymbol{d}\right)}{\left|b_{k,1}(\boldsymbol{d})\right|}\right)\right], & \text{linear} \\ \dfrac{1-b_{k,2}(\boldsymbol{d})/\left|b_{k,2}(\boldsymbol{d})\right|}{2} + \mathbb{E}\left[\Phi\left(\dfrac{-b_{k,1}(\boldsymbol{d})+\sqrt{b_{k,1}^2(\boldsymbol{d})-4b_{k,2}(\boldsymbol{d})\left[\beta_k(\boldsymbol{d})+B\left(\widetilde{\boldsymbol{V}}_k;\boldsymbol{d}\right)\right]}}{2b_{k,2}(\boldsymbol{d})}\right)\right], & \text{quadratic} \\ \quad -\mathbb{E}\left[\Phi\left(\dfrac{-b_{k,1}(\boldsymbol{d})-\sqrt{b_{k,1}^2(\boldsymbol{d})-4b_{k,2}(\boldsymbol{d})\left[\beta_k(\boldsymbol{d})+B\left(\widetilde{\boldsymbol{V}}_k;\boldsymbol{d}\right)\right]}}{2b_{k,2}(\boldsymbol{d})}\right)\right] \end{cases} \tag{5}$$

where $\Phi(z) = \left(1/\sqrt{2\pi}\right)\int_{-\infty}^{z}\exp\left(-\xi^2/2\right)d\xi$ is the cumulative distribution function of a standard Gaussian random variable, $\widetilde{\boldsymbol{V}}_k = \left\{V_{k,1},\cdots,V_{k,N-1}\right\}^T$ is an N-1-dimensional standard Gaussian random vector, and $B\left(\widetilde{\boldsymbol{V}}_k;\boldsymbol{d}\right) \equiv b_{k,0}(\boldsymbol{d}) + \sum_{i=1}^{N-1} y_{k,i}\left(V_{k,i};\boldsymbol{d}\right)$. Note that both expressions in (5) provide higher-order estimates of the failure probability than those provided by FORM/SORM if univariate component functions $y_{k,i}\left(v_{k,i};\boldsymbol{d}\right), i=1,N-1$ are approximated by higher than second-order terms. Integrating with respect to $\widetilde{\boldsymbol{v}}_k = \left\{v_{k,1},\cdots,v_{k,N-1}\right\}^T \in \mathbb{R}^{N-1}$ yields

$$c_k(\boldsymbol{d}) \cong \begin{cases} \int_{\mathbb{R}^{N-1}}\Phi\left(\dfrac{-\beta_k(\boldsymbol{d})-b_{k,0}(\boldsymbol{d})-\sum_{i=1}^{N-1}y_{k,i}(v_{k,i};\boldsymbol{d})}{\left|b_{k,1}(\boldsymbol{d})\right|}\right)\prod_{i=1}^{N-1}\phi(\boldsymbol{v}_{k,i})d\boldsymbol{v}_{k,i}, & \text{linear} \\ \dfrac{1-b_{k,2}(\boldsymbol{d})/\left|b_{k,2}(\boldsymbol{d})\right|}{2} + \int_{\mathbb{R}^{N-1}}\Phi\left(\dfrac{-b_{k,1}(\boldsymbol{d})+\sqrt{b_{k,1}^2(\boldsymbol{d})-4b_{k,2}(\boldsymbol{d})[\beta_k(\boldsymbol{d})+B(\widetilde{\boldsymbol{v}}_k;\boldsymbol{d})]}}{2b_{k,2}(\boldsymbol{d})}\right)\prod_{i=1}^{N-1}\phi(v_{k,i})d\boldsymbol{v}_{k,i}, & \text{quadratic} \\ \quad -\int_{\mathbb{R}^{N-1}}\Phi\left(\dfrac{-b_{k,1}(\boldsymbol{d})-\sqrt{b_{k,1}^2(\boldsymbol{d})-4b_{k,2}(\boldsymbol{d})[\beta_k(\boldsymbol{d})+B(\widetilde{\boldsymbol{v}}_k;\boldsymbol{d})]}}{2b_{k,2}(\boldsymbol{d})}\right)\prod_{i=1}^{N-1}\phi\left(\boldsymbol{v}_{k,i}\right)d\boldsymbol{v}_{k,i} \end{cases} \tag{6}$$

where $\phi\left(v_{k,i}\right) \equiv d\Phi\left(v_{k,i}\right)/dv_{k,i} = \left(1/\sqrt{2\pi}\right)\exp\left(-v_{k,i}^2/2\right)$ is the probability density function of a standard Gaussian random variable. Regardless of the linear or quadratic approximations, the constraint evaluation requires performing multidimensional integrations over $\mathbb{R}^{N-1}$.

[1] Compared with the work of Wei and Rahman (2007), (5) contains the term $\beta_k(\boldsymbol{d})$ as a part of b_0 and does not include the term $y_k\left(\boldsymbol{v}_k^*;\boldsymbol{d}\right)$, which is zero.

3.2 Design sensitivity analysis

In gradient-based optimization algorithms, derivatives of both objective and constraint functions with respect to each design variable are required. For problem $\mathcal{P}_2$, calculating derivatives of the objective function is trivial. However, the formulation of gradients for a constraint function is dependent on how the underlying reliability analysis is performed. A new, analytically derived sensitivity analysis of general constraint functions $c_k(\boldsymbol{d}); k = 1, K$ was conducted as follows.

The integrand of the multi-dimensional integration (6) depends on $\beta_k(\boldsymbol{d})$ and $\widetilde{\boldsymbol{v}}_k(\boldsymbol{d})$, each of which in turn depends on design $\boldsymbol{d}$. By applying the chain rule in (6), the partial derivative of the constraint function $c_k(\boldsymbol{d})$ with respect to a design variable d_i is

$$\frac{\partial c_k(\boldsymbol{d})}{\partial d_i} = \frac{\partial c_k(\boldsymbol{d})}{\partial \beta_k}\frac{\partial \beta_k}{\partial d_i} + \sum_{j=1}^{N-1}\frac{\partial c_k(\boldsymbol{d})}{\partial v_{k,j}}\frac{\partial v_{k,j}}{\partial d_i} \tag{7}$$

which involves four partial derivatives in the right-hand side, described as follows. Like reliability analysis, (7) also provides higher-order approximations of design sensitivities than those calculated from FORM/SORM.

3.2.1 Partial derivative of $c_k(d)$ with respect to β_k

The evaluation of $\partial c_k(\boldsymbol{d})/\partial \beta_k$ can be simplified by assuming that the coefficients $b_{k,0}(\boldsymbol{d})$, $b_{k,1}(\boldsymbol{d})$, $b_{k,2}(\boldsymbol{d})$ of $y_{k,N}\left(v_{k,N};\boldsymbol{d}\right)$, univariate functions $y_{k,i}\left(v_{k,i};d\right); i = 1, N-1$, and hence, $B(\widetilde{\boldsymbol{v}}_k;\boldsymbol{d})$ do not change significantly with β_k. With the assumption of their constancy at a given design,

$$\frac{\partial c_k(\boldsymbol{d})}{\partial \beta_k} \cong \int_{\mathbb{R}^{N-1}} -\phi\left(\frac{-\beta_k(\boldsymbol{d}) - b_{k,0}(\boldsymbol{d}) - \sum_{i=1}^{N-1} y_{k,i}(v_{k,i};\boldsymbol{d})}{|b_{k,1}(\boldsymbol{d})|}\right)\frac{1}{|b_{k,1}(\boldsymbol{d})|}\prod_{i=1}^{N-1}\phi\left(v_{k,i}\right)dv_{k,i} \tag{8}$$

and

$$\frac{\partial c_k(\boldsymbol{d})}{\partial \beta_k} \cong \int_{\mathbb{R}^{N-1}} -\left[\phi\left(\frac{-b_{k,1}(\boldsymbol{d}) + \sqrt{b_{k,1}^2(\boldsymbol{d}) - 4b_{k,2}(\boldsymbol{d})[\beta_k(\boldsymbol{d}) + B(\widetilde{\boldsymbol{v}}_k;\boldsymbol{d})]}}{2b_{k,2}(\boldsymbol{d})}\right) + \phi\left(\frac{-b_{k,1}(\boldsymbol{d}) - \sqrt{b_{k,1}^2(\boldsymbol{d}) - 4b_{k,2}(\boldsymbol{d})[\beta_k(\boldsymbol{d}) + B(\widetilde{\boldsymbol{v}}_k;\boldsymbol{d})]}}{2b_{k,2}(\boldsymbol{d})}\right)\right] \times \frac{1}{\sqrt{b_{k,1}^2(\boldsymbol{d}) - 4b_{k,2}(\boldsymbol{d})[\beta_k(\boldsymbol{d}) + B(\widetilde{\boldsymbol{v}}_k;\boldsymbol{d})]}}\prod_{i=1}^{N-1}\phi\left(v_{k,i}\right)dv_{k,i}, \tag{9}$$

for linear and quadratic approximations, respectively, of $y_{k,N}\left(v_{k,N};\boldsymbol{d}\right)$. Both (8) and (9) require evaluating multivariate integrals over $\mathbb{R}^{N-1}$.

3.2.2 Partial derivative of $c_k(d)$ with respect to $v_{k,\,j}$

Using linear and quadratic approximations of $y_{k,N}\left(v_{k,N};\boldsymbol{d}\right)$, the partial derivatives of the failure probability defined by the constraint function $c_k(\boldsymbol{d})$ with respect to $v_{k,j}$ are

$$\frac{\partial c_k(\boldsymbol{d})}{\partial v_{k,j}} \cong \int_{\mathbb{R}^{N-1}} \left[-\phi\left(\frac{-\beta_k(\boldsymbol{d}) - b_{k,0}(\boldsymbol{d}) - \sum_{i=1}^{N-1} y_{k,i}\left(v_{k,i};\boldsymbol{d}\right)}{|b_{k,1}(d)|}\right)\frac{\frac{\partial y_{k,j}\left(v_{k,j};\boldsymbol{d}\right)}{\partial v_{k,j}}}{|b_{k,1}(\boldsymbol{d})|} - \Phi\left(\frac{-\beta_k(\boldsymbol{d}) - b_{k,0}(\boldsymbol{d}) - \sum_{i=1}^{N-1} y_{k,i}\left(v_{k,i};\boldsymbol{d}\right)}{|b_{k,1}(\boldsymbol{d})|}\right)v_{k,j}\right]\prod_{i=1}^{N-1}\phi\left(v_{k,i}\right)dv_{k,i} \tag{10}$$

$$\frac{\partial c_k(\boldsymbol{d})}{\partial v_{k,j}} \cong \int_{\mathbb{R}^{N-1}} -\left[\phi\left(\frac{-b_{k,1}(\boldsymbol{d}) + \sqrt{b_{k,1}^2(\boldsymbol{d}) - 4b_{k,2}(\boldsymbol{d})[\beta_k(\boldsymbol{d}) + B(\widetilde{\boldsymbol{v}}_k;\boldsymbol{d})]}}{2b_{k,2}(\boldsymbol{d}))}\right)\frac{\frac{\partial y_{k,j}\left(v_{k,j};\boldsymbol{d}\right)}{\partial v_{k,j}}}{\sqrt{b_{k,1}^2(\boldsymbol{d}) - 4b_{k,2}(\boldsymbol{d})[\beta_k(\boldsymbol{d}) + B(\widetilde{\boldsymbol{v}}_k;\boldsymbol{d})]}} + \Phi\left(\frac{-b_{k,1}(\boldsymbol{d}) + \sqrt{b_{k,1}^2(\boldsymbol{d}) - 4b_{k,2}(\boldsymbol{d})[\beta_k(\boldsymbol{d}) + B(\widetilde{\boldsymbol{v}}_k;\boldsymbol{d})]}}{2b_{k,2}(\boldsymbol{d})}\right)\right]v_{k,j}\prod_{i=1}^{N-1}\phi\left(v_{k,i}\right)dv_{k,i} \int_{\mathbb{R}^{N-1}}\left[\phi\left(\frac{-b_{k,1}(\boldsymbol{d}) - \sqrt{b_{k,1}^2(\boldsymbol{d}) - 4b_{k,2}(\boldsymbol{d})[\beta_k(\boldsymbol{d}) + B(\widetilde{\boldsymbol{v}}_k;\boldsymbol{d})]}}{2b_{k,2}(\boldsymbol{d})}\right)\frac{\frac{\partial y_{k,j}\left(v_{k,j};\boldsymbol{d}\right)}{\partial v_{k,j}}}{\sqrt{b_{k,1}^2(\boldsymbol{d}) - 4b_{k,2}(\boldsymbol{d})[\beta_k(\boldsymbol{d}) + B(\widetilde{\boldsymbol{v}}_k;\boldsymbol{d})]}} - \Phi\left(\frac{-b_{k,1}(\boldsymbol{d}) - \sqrt{b_{k,1}^2(\boldsymbol{d}) - 4b_{k,2}(\boldsymbol{d})[\beta_k(\boldsymbol{d}) + B(\widetilde{\boldsymbol{v}}_k;\boldsymbol{d})]}}{2b_{k,2}(\boldsymbol{d})}\right)v_{k,j}\right]\prod_{i=1}^{N-1}\phi\left(v_{k,i}\right)dv_{k,i} \tag{11}$$

respectively, where $\partial y_{k,j}\left(v_{k,j};\boldsymbol{d}\right)/\partial v_{k,j}$ can be calculated directly or by taking the partial derivative of the Lagrange interpolation of $y_{k,N}\left(v_{k,N};\boldsymbol{d}\right)$ (Rahman and Wei 2006). Again, both (10) and (11) involve multivariate integration over $\mathbb{R}^{N-1}$.

3.2.3 Partial derivatives of β_k and $v_{k,j}$ with respect to d_i

Of the two remaining gradients, the partial derivative of the reliability index $\beta_{k,}$ with respect to design variable, d_i is (Ditlevsen and Madsen 1996)

$$\frac{\partial \beta_k}{\partial d_i} = -\left[\frac{\nabla h_k(\boldsymbol{u};\boldsymbol{d})^T}{\|\nabla h_k(\boldsymbol{u};\boldsymbol{d})\|}\frac{\partial \boldsymbol{u}}{\partial d_i}\right]_{\boldsymbol{u}=\boldsymbol{u}_k^*} \tag{12}$$

where $\nabla h_k = \{\partial h_k/\partial u_1, \cdots, \partial h_k/\partial u_N\}^T$, $\|\cdot\|$ is the $\mathbb{L}_2$ norm, and the vector derivative $\partial \boldsymbol{u}/\partial d_i = \{\partial u_1/\partial d_i, \cdots,$

$\partial u_N/\partial d_i\}^T$ is obtained from the $\boldsymbol{x}$–$\boldsymbol{u}$ transformation. In (12), $\nabla h_k(u;d)^T$ represents a vector of structural response sensitivities, is problem dependent, and is calculated either analytically or numerically by a finite-difference approximation. Finally, the partial derivative $\partial v_{k,j}/\partial d_i$ included in (7) is obtained from the $\boldsymbol{x}$–$\boldsymbol{v}_k$ transformation. Both $\boldsymbol{x}$–$\boldsymbol{u}$ and $\boldsymbol{x}$–$\boldsymbol{v}_k$ transformations depend on the probability distribution of $\boldsymbol{X}$ and, hence, on a specific RBDO problem to be solved.

3.3 Univariate numerical integration for reliability and sensitivity analyses

The expressions of the constraint function in (6) and partial derivatives in (8), (9), (10), and (11) involve multivariate integrations over $\mathbb{R}^{N-1}$. A generic evaluation of these integrals requires calculating $\int_{\mathbb{R}^{N-1}} f_k(\widetilde{\boldsymbol{v}}_k)\prod_{i=1}^{N-1}\phi(v_{k,i})dv_{k,i}$, where $f_k:\mathbb{R}^{N-1}\to\mathbb{R}$ is the multivariate part of the integrand, is non-negative, and depends on how univariate component functions $y_{k,N}(v_{k,N};\boldsymbol{d})$ are constructed. The exact calculation of this integral is not possible in general. Direct numerical integration is not efficient when the dimension exceeds three or four.

In reference to (3), consider again a univariate approximation of $\ln[f_k(\widetilde{\boldsymbol{v}}_k)]$ at $\widetilde{\boldsymbol{v}}_k^* \equiv \{0,\cdots,0\}^T = \boldsymbol{0}\in\mathbb{R}^{N-1}$, expressed by

$$\ln[f_k(\widetilde{\boldsymbol{v}}_k)] \cong \sum_{i=1}^{N-1}\ln\left[f_{k,i}(v_{k,i})\right] - (N-2)\ln[f_k(\boldsymbol{0})] \tag{13}$$

where $f_{k,i}(v_{k,i}) \equiv f_k(0,\cdots,0,v_{k,i},0,\cdots,0)$ are univariate component functions of $f_k(\widetilde{\boldsymbol{v}}_k)$ and $f_k(\boldsymbol{0}) \equiv f_k(0,\cdots,0)$. Hence,

$$\begin{aligned} f_k(\widetilde{\boldsymbol{v}}_k) &= \exp\{\ln[f_k(\widetilde{\boldsymbol{v}}_k)]\} \\ &\cong \exp\left\{\sum_{i=1}^{N-1}\ln\left[f_{k,i}(v_{k,i})\right] - (N-2)\ln[f_k(\boldsymbol{0})]\right\} \\ &= \frac{\prod_{i=1}^{N-1} f_{k,i}(v_{k,i})}{f_k(\boldsymbol{0})^{N-2}} \end{aligned} \tag{14}$$

yielding

$$\begin{aligned} &\int_{\mathbb{R}^{N-1}} f_k(\widetilde{\boldsymbol{v}}_k)\prod_{i=1}^{N-1}\phi(v_{k,i})dv_{k,i} \\ &\cong \frac{1}{f_k(\boldsymbol{0})^{N-2}}\prod_{i=1}^{N-1}\int_{-\infty}^{+\infty} f_{k,i}(v_{k,i})\phi(v_{k,i})dv_{k,i}, \end{aligned} \tag{15}$$

which involves a product of N-1 *univariate* integrals. Using (15) with appropriately defined $f_k(\widetilde{\boldsymbol{v}}_k)$ in (6) and (8), (9), (10), and (11), the failure probability and their derivatives are as follows.

3.3.1 Linear approximation

$$c_k(\boldsymbol{d}) \cong \frac{\prod_{i=1}^{N-1}\int_{-\infty}^{+\infty}\Phi\left(\frac{-\beta_k(\boldsymbol{d})-b_{k,0}(\boldsymbol{d})-y_{k,i}(v_{k,i};\boldsymbol{d})}{|b_{k,1}(\boldsymbol{d})|}\right)\phi(v_{k,i})dv_{k,i}}{\left[\Phi\left(\frac{-\beta_k(\boldsymbol{d})-b_{k,0}(\boldsymbol{d})}{|b_{k,1}(\boldsymbol{d})|}\right)\right]^{N-2}} \tag{16}$$

$$\frac{\partial c_k(\boldsymbol{d})}{\partial\beta_k} \cong -\frac{\prod_{i=1}^{N-1}\int_{-\infty}^{+\infty}\phi\left(\frac{-\beta_k(\boldsymbol{d})-b_{k,0}(\boldsymbol{d})-y_{k,i}(v_{k,i};\boldsymbol{d})}{|b_{k,1}(\boldsymbol{d})|}\right)\phi(v_{k,i})dv_{k,i}}{|b_{k,1}(\boldsymbol{d})|\left[\phi\left(\frac{-\beta_k(\boldsymbol{d})-b_{k,0}(\boldsymbol{d})}{|b_{k,1}(\boldsymbol{d})|}\right)\right]^{N-2}} \tag{17}$$

$$\begin{aligned} \frac{\partial c_k(d)}{\partial v_{k,j}} \cong &-\frac{\prod_{i=1,i\neq j}^{N-1}\int_{-\infty}^{+\infty}\phi\left(\frac{-\beta_k(\boldsymbol{d})-b_{k,0}(\boldsymbol{d})-y_{k,i}(v_{k,i};\boldsymbol{d})}{|b_{k,1}(\boldsymbol{d})|}\right)\phi(v_{k,i})dv_{k,i}}{|b_{k,1}(\boldsymbol{d})|\left[\phi\left(\frac{-\beta_k(\boldsymbol{d})-b_{k,0}(\boldsymbol{d})}{|b_{k,1}(\boldsymbol{d})|}\right)\right]^{N-2}}\times \\ &\int_{-\infty}^{+\infty}\phi\left(\frac{-\beta_k(\boldsymbol{d})-b_{k,0}(d)-y_{k,j}(v_{k,j};d)}{|b_{k,1}(\boldsymbol{d})|}\right)\frac{\partial y_{k,j}(v_{k,j};\boldsymbol{d})}{\partial v_{k,j}}\phi(v_{k,j})dv_{k,j}- \\ &\frac{\prod_{i=1,i\neq j}^{N-1}\int_{-\infty}^{+\infty}\Phi\left(\frac{-\beta_k(\boldsymbol{d})-b_{k,0}(\boldsymbol{d})-y_{k,i}(v_{k,i};\boldsymbol{d})}{|b_{k,1}(\boldsymbol{d})|}\right)\phi(v_{k,i})dv_{k,i}}{\left[\Phi\left(\frac{-\beta_k(\boldsymbol{d})-b_{k,0}(\boldsymbol{d})}{|b_{k,1}(\boldsymbol{d})|}\right)\right]^{N-2}}\times \\ &\int_{-\infty}^{+\infty}\Phi\left(\frac{-\beta_k(\boldsymbol{d})-b_{k,0}(\boldsymbol{d})-y_{k,j}(v_{k,j};\boldsymbol{d})}{|b_{k,1}(\boldsymbol{d})|}\right)v_{k,j}\phi(v_{k,j})dv_{k,j} \end{aligned} \tag{18}$$

3.3.2 Quadratic approximation

$$c_k(\boldsymbol{d}) \cong \frac{1-b_{k,2}(\boldsymbol{d})/|b_{k,2}(d)|}{2} + I_1(\boldsymbol{d}) + I_2(\boldsymbol{d}) \tag{19}$$

$$\frac{\partial c_k(\boldsymbol{d})}{\partial\beta_k} \cong I_3(\boldsymbol{d}) + I_4(\boldsymbol{d}), \tag{20}$$

$$\frac{\partial c_k(\boldsymbol{d})}{\partial v_{k,j}} \cong I_5(\boldsymbol{d}) + I_6(\boldsymbol{d}) + I_7(\boldsymbol{d}) + I_8(\boldsymbol{d}), \tag{21}$$

where

$$I_1(\boldsymbol{d}) = \frac{\prod_{i=1}^{N-1}\int_{-\infty}^{+\infty}\Phi\left(\frac{-b_{k,1}(\boldsymbol{d})+\sqrt{b_{k,1}^2(\boldsymbol{d})-4b_{k,2}(\boldsymbol{d})\left[\beta_k(\boldsymbol{d})+B_i\left(v_{k,i};\boldsymbol{d}\right)\right]}}{2b_{k,2}(\boldsymbol{d})}\right)\Phi\left(v_{k,i}\right)dv_{k,i}}{\left[\Phi\left(\frac{-b_{k,1}(\boldsymbol{d})+\sqrt{b_{k,1}^2(\boldsymbol{d})-4b_{k,2}(\boldsymbol{d})[\beta_k(\boldsymbol{d})+B(\boldsymbol{0};\boldsymbol{d})]}}{2b_{k,2}(\boldsymbol{d})}\right)\right]^{N-2}}, \tag{22}$$

$$I_2(\boldsymbol{d}) = -\frac{\prod_{i=1}^{N-1}\int_{-\infty}^{+\infty}\Phi\left(\frac{-b_{k,1}(\boldsymbol{d})-\sqrt{b_{k,1}^2(\boldsymbol{d})-4b_{k,2}(\boldsymbol{d})\left[\beta_k(\boldsymbol{d})+B_i\left(v_{k,i};\boldsymbol{d}\right)\right]}}{2b_{k,2}(\boldsymbol{d})}\right)\Phi\left(v_{k,i}\right)dv_{k,i}}{\left[\Phi\left(\frac{-b_{k,1}(\boldsymbol{d})-\sqrt{b_{k,1}^2(\boldsymbol{d})-4b_{k,2}(\boldsymbol{d})[\beta_k(\boldsymbol{d})+B(\boldsymbol{0};\boldsymbol{d})]}}{2b_{k,2}(\boldsymbol{d})}\right)\right]^{N-2}}, \tag{23}$$

$$I_3(\boldsymbol{d}) \cong -\prod_{i=1}^{N-1}\int_{-\infty}^{+\infty}\frac{\Phi\left(\frac{-b_{k,1}(\boldsymbol{d})+\sqrt{b_{k,1}^2(\boldsymbol{d})-4b_{k,2}(\boldsymbol{d})\left[\beta_k(\boldsymbol{d})+B_i\left(v_{k,i};\boldsymbol{d}\right)\right]}}{2b_{k,2}(\boldsymbol{d})}\right)}{\sqrt{b_{k,1}^2(\boldsymbol{d})-4b_{k,2}(\boldsymbol{d})\left[\beta_k(\boldsymbol{d})+B_i\left(v_{k,i};\boldsymbol{d}\right)\right]}}\Phi\left(v_{k,i}\right)dv_{k,i}\times \left[\frac{\Phi\left(\frac{-b_{k,1}(\boldsymbol{d})+\sqrt{b_{k,1}^2(\boldsymbol{d})-4b_{k,2}(\boldsymbol{d})\left[\beta_k(\boldsymbol{d})+B(\boldsymbol{0};\boldsymbol{d})\right]}}{2b_{k,2}(\boldsymbol{d})}\right)}{\sqrt{b_{k,1}^2(\boldsymbol{d})-4b_{k,2}(\boldsymbol{d})\left[\beta_k(\boldsymbol{d})+B(\boldsymbol{0};\boldsymbol{d})\right]}}\right]^{-(N-2)}, \tag{24}$$

$$I_4(\boldsymbol{d}) \cong -\prod_{i=1}^{N-1}\int_{-\infty}^{+\infty}\frac{\Phi\left(\frac{-b_{k,1}(\boldsymbol{d})-\sqrt{b_{k,1}^2(\boldsymbol{d})-4b_{k,2}(\boldsymbol{d})\left[\beta_k(\boldsymbol{d})+B_i\left(v_{k,i};\boldsymbol{d}\right)\right]}}{2b_{k,2}(\boldsymbol{d})}\right)}{\sqrt{b_{k,1}^2(\boldsymbol{d})-4b_{k,2}(\boldsymbol{d})\left[\beta_k(\boldsymbol{d})+B_i\left(v_{k,i};\boldsymbol{d}\right)\right]}}\Phi\left(v_{k,i}\right)dv_{k,i}\times \left[\frac{\Phi\left(\frac{-b_{k,1}(\boldsymbol{d})-\sqrt{b_{k,1}^2(\boldsymbol{d})-4b_{k,2}(\boldsymbol{d})\left[\beta_k(\boldsymbol{d})+B(\boldsymbol{0};\boldsymbol{d})\right]}}{2b_{k,2}(\boldsymbol{d})}\right)}{\sqrt{b_{k,1}^2(\boldsymbol{d})-4b_{k,2}(\boldsymbol{d})\left[\beta_k(\boldsymbol{d})+B(\boldsymbol{0};\boldsymbol{d})\right]}}\right]^{-(N-2)}, \tag{25}$$

$$I_5(\boldsymbol{d}) = -\prod_{i=1,i\neq j}^{N-1}\int_{-\infty}^{+\infty}\frac{\Phi\left(\frac{-b_{k,1}(\boldsymbol{d})+\sqrt{b_{k,1}^2(\boldsymbol{d})-4b_{k,2}(\boldsymbol{d})\left[\beta_k(\boldsymbol{d})+B_i\left(v_{k,i};\boldsymbol{d}\right)\right]}}{2b_{k,2}(\boldsymbol{d})}\right)}{\sqrt{b_{k,1}^2(\boldsymbol{d})-4b_{k,2}(\boldsymbol{d})\left[\beta_k(\boldsymbol{d})+B_i\left(v_{k,i};\boldsymbol{d}\right)\right]}}\Phi\left(v_{k,i}\right)dv_{k,i}\times \int_{-\infty}^{+\infty}\frac{\Phi\left(\frac{-b_{k,1}(\boldsymbol{d})+\sqrt{b_{k,1}^2(\boldsymbol{d})-4b_{k,2}(\boldsymbol{d})\left[\beta_k(\boldsymbol{d})+B_j\left(v_{k,j};\boldsymbol{d}\right)\right]}}{2b_{k,2}(\boldsymbol{d})}\right)}{\sqrt{b_{k,1}^2(\boldsymbol{d})-4b_{k,2}(\boldsymbol{d})\left[\beta_k(\boldsymbol{d})+B_j\left(v_{k,j};\boldsymbol{d}\right)\right]}}\frac{\partial y_{k,j}\left(v_{k,j};\boldsymbol{d}\right)}{\partial v_{k,j}}\Phi\left(v_{k,j}\right)dv_{k,j}\times \left[\frac{\Phi\left(\frac{-b_{k,1}(\boldsymbol{d})+\sqrt{b_{k,1}^2(\boldsymbol{d})-4b_{k,2}(\boldsymbol{d})\left[\beta_k(\boldsymbol{d})+B(\boldsymbol{0};\boldsymbol{d})\right]}}{2b_{k,2}(\boldsymbol{d})}\right)}{\sqrt{b_{k,1}^2(\boldsymbol{d})-4b_{k,2}(\boldsymbol{d})\left[\beta_k(d)+B(\boldsymbol{0};\boldsymbol{d})\right]}}\right]^{-(N-2)} \tag{26}$$

$$I_6(\boldsymbol{d}) = -\prod_{i=1,i\neq j}^{N-1}\int_{-\infty}^{+\infty}\Phi\left(\frac{-b_{k,1}(\boldsymbol{d})+\sqrt{b_{k,1}^2(\boldsymbol{d})-4b_{k,2}(\boldsymbol{d})\left[\beta_k(\boldsymbol{d})+B_i\left(v_{k,i};\boldsymbol{d}\right)\right]}}{2b_{k,2}(\boldsymbol{d})}\right)\Phi\left(v_{k,i}\right)dv_{k,i}\times \int_{-\infty}^{+\infty}\Phi\left(\frac{-b_{k,1}(\boldsymbol{d})+\sqrt{b_{k,1}^2(\boldsymbol{d})-4b_{k,2}(\boldsymbol{d})\left[\beta_k(\boldsymbol{d})+B_j\left(v_{k,j};\boldsymbol{d}\right)\right]}}{2b_{k,2}(\boldsymbol{d})}\right)v_{k,j}\Phi\left(v_{k,j}\right)dv_{k,j}\times, \left[\Phi\left(\frac{-b_{k,1}(\boldsymbol{d})+\sqrt{b_{k,1}^2(\boldsymbol{d})-4b_{k,2}(\boldsymbol{d})[\beta_k(\boldsymbol{d})+B(\boldsymbol{0};\boldsymbol{d})]}}{2b_{k,2}(\boldsymbol{d})}\right)\right]^{-(N-2)} \tag{27}$$

$$I_7(\boldsymbol{d}) = -\prod_{i=1,i\neq j}^{N-1}\int_{-\infty}^{+\infty}\frac{\Phi\left(\frac{-b_{k,1}(\boldsymbol{d})-\sqrt{b_{k,1}^2(\boldsymbol{d})-4b_{k,2}(\boldsymbol{d})\left[\beta_k(\boldsymbol{d})+B_i\left(v_{k,i};\boldsymbol{d}\right)\right]}}{2b_{k,2}(\boldsymbol{d})}\right)}{\sqrt{b_{k,1}^2(\boldsymbol{d})-4b_{k,2}(\boldsymbol{d})\left[\beta_k(\boldsymbol{d})+B_i\left(v_{k,i};\boldsymbol{d}\right)\right]}}\Phi\left(v_{k,i}\right)dv_{k,i}\times \int_{-\infty}^{+\infty}\frac{\Phi\left(\frac{-b_{k,1}(\boldsymbol{d})-\sqrt{b_{k,1}^2(\boldsymbol{d})-4b_{k,2}(\boldsymbol{d})\left[\beta_k(\boldsymbol{d})+B_j\left(v_{k,j};\boldsymbol{d}\right)\right]}}{2b_{k,2}(\boldsymbol{d})}\right)}{\sqrt{b_{k,1}^2(\boldsymbol{d})-4b_{k,2}(\boldsymbol{d})\left[\beta_k(\boldsymbol{d})+B_j\left(v_{k,j};\boldsymbol{d}\right)\right]}}\frac{\partial y_{k,j}\left(v_{k,j};\boldsymbol{d}\right)}{\partial v_{k,j}}\Phi\left(v_{k,j}\right)dv_{k,j}\times, \left[\frac{\Phi\left(\frac{-b_{k,1}(\boldsymbol{d})-\sqrt{b_{k,1}^2(\boldsymbol{d})-4b_{k,2}(\boldsymbol{d})\left[\beta_k(\boldsymbol{d})+B(\boldsymbol{0};\boldsymbol{d})\right]}}{2b_{k,2}(\boldsymbol{d})}\right)}{\sqrt{b_{k,1}^2(\boldsymbol{d})-4b_{k,2}(\boldsymbol{d})\left[\beta_k(\boldsymbol{d})+B(\boldsymbol{0};\boldsymbol{d})\right]}}\right]^{-(N-2)} \tag{28}$$

$$I_8(\boldsymbol{d}) = \prod_{i=1,i\neq j}^{N-1}\int_{-\infty}^{+\infty}\Phi\left(\frac{-b_{k,1}(\boldsymbol{d})-\sqrt{b_{k,1}^2(\boldsymbol{d})-4b_{k,2}(\boldsymbol{d})\left[\beta_k(\boldsymbol{d})+B_i\left(v_{k,i};\boldsymbol{d}\right)\right]}}{2b_{k,2}(d)}\right)\Phi\left(v_{k,i}\right)dv_{k,i}\times \int_{-\infty}^{+\infty}\Phi\left(\frac{-b_{k,1}(\boldsymbol{d})-\sqrt{b_{k,1}^2(\boldsymbol{d})-4b_{k,2}(\boldsymbol{d})\left[\beta_k(\boldsymbol{d})+B_j\left(v_{k,j};\boldsymbol{d}\right)\right]}}{2b_{k,2}(\boldsymbol{d})}\right)v_{k,j}\Phi\left(v_{k,j}\right)dv_{k,j}\times, \left[\Phi\left(\frac{-b_{k,1}(\boldsymbol{d})-\sqrt{b_{k,1}^2(\boldsymbol{d})-4b_{k,2}(d)[\beta_k(\boldsymbol{d})+B(\boldsymbol{0};\boldsymbol{d})]}}{2b_{k,2}(\boldsymbol{d})}\right)\right]^{-(N-2)} \tag{29}$$

and $B_i\left(v_{k,i};\boldsymbol{d}\right) \equiv B\left(0,\cdots,0,v_{k,i},0,\cdots,0;\boldsymbol{d}\right)$. After determining the coefficients $b_{k,0}(\boldsymbol{d}), b_{k,1}(\boldsymbol{d})$, and $b_{k,2}(\boldsymbol{d})$, the reliability index $\beta_k(\boldsymbol{d})$, and the univariate component functions $y_{k,i}\left(v_{k,i};\boldsymbol{d}\right); i = 1, N-1$, the univariate integration involved in (16), (17), (18), (22), (23), (24), (25), (26), (27), (28), and (29) can be easily evaluated by one-dimensional Gauss–Hermite quadrature (Abramowitz and Stegun 1972). Equation (7), with partial derivatives formulated in (17), (18), (20), and (21), provides design sensitivities for gradient-based design optimization.

Equations (16), (17), and (18) represent generalized expressions that can be degenerated to existing FORM/SORM equations by selecting appropriate values of $b_{k,0}(\boldsymbol{d})$,

$b_{k,1}(\boldsymbol{d})$ and univariate component functions $y_{k,i}(v_{k,i};\boldsymbol{d})$; $i=1,N-1$. For example, the selection of $b_{k,0}(\boldsymbol{d})=0$, $b_{k,1}(\boldsymbol{d})=-1$ and $y_{k,i}(v_{k,i};\boldsymbol{d})=0$; $i=1,N-1$ in (16) (17), and (18) leads to $c_k(\boldsymbol{d})=\Phi(-\beta_k(\boldsymbol{d}))$ and $\partial c_k(\boldsymbol{d})/\partial d_i=-\Phi(-\beta_k(\boldsymbol{d}))\partial\beta_k(\boldsymbol{d})/\partial d_i$, which are the well-known first-order (FORM) approximations of the failure probability and design sensitivity, respectively. A similar reduction process can be employed to derive the second-order (SORM) approximations. In contrast, (16), (17), and (18) provide failure probability and design sensitivity estimates of an arbitrarily high order, where the order is determined by the selection of univariate component functions $y_{k,i}(v_{k,i};\boldsymbol{d})=0$; $i=1,N-1$. Therefore, (16), (17), and (18) are capable of providing estimates of $c_k(\boldsymbol{d})$ and $\partial c_k(\boldsymbol{d})/\partial d_i$ that are superior to those provided by commonly used FORM/SORM approximations. More importantly, the design sensitivities are calculated without incurring function evaluations in addition to those required for reliability analysis. It is worth noting that, unlike (16), (17), and (18), (19), (20), and (21) cannot be reduced to FORM/SORM equations, as $y_{k,N}(v_{k,N};\boldsymbol{d})$ contains a second-order term.

3.4 Computational flow and effort

In summary, the overall process for solving the RBDO problem $\mathcal{P}_2$ can be described by the following steps:

1. Define an initial design with $\boldsymbol{d}=\boldsymbol{d}_0$. Use the final result of mean-based or other relevant reference-point-based optimization if available.
2. Evaluate both objective and constraint functions for the current design vector. For constraint functions, use the proposed univariate decomposition method (16 or 19) for reliability analysis.
3. Evaluate the gradients of both the objective and constraint functions for the current design vector. For gradients of constraint functions, use the proposed univariate decomposition method (7, 17 and 18, or 7, 20, and 21) for design sensitivity analysis.
4. Perform deterministic optimization to solve (2) by a selected gradient-based algorithm.
5. Check for the convergence of the objective function and design vector. If the convergence is reached, stop. If not, update the design vector to find the next design vector and repeat steps 2 through 4.

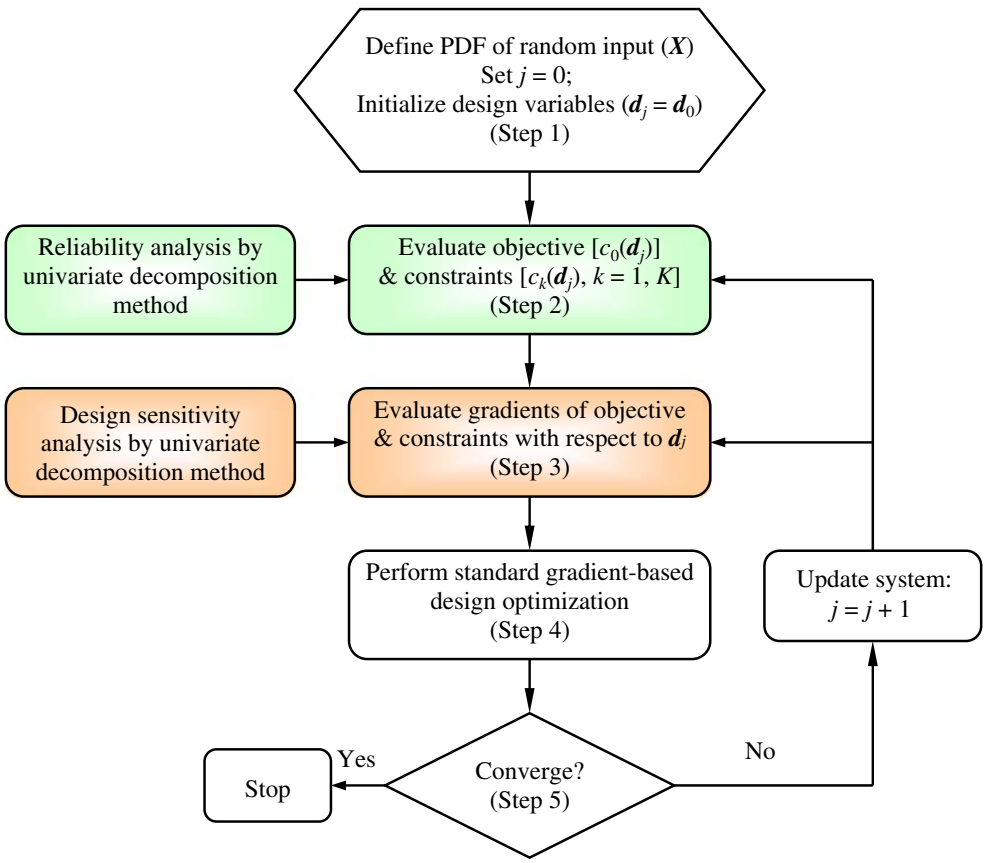

Fig. 2 Flowchart of the proposed RBDO process

Figure 2 depicts the flowchart of the proposed RBDO process. New methods were developed in the shaded areas.

For determining computational effort, consider $y_{k,i}\left(v_{k,i};\boldsymbol{d}\right) \equiv y_k\left(0,\cdots,0,v_{k,i},0,\cdots,\beta_k;\boldsymbol{d}\right); i=1,N,$ for which n function values $y_{k,i}\left(v_{k,i}^{(j)};\boldsymbol{d}\right) \equiv y\left(0,\cdots,0,v_{k,i}^{(j)},0,\cdots,\beta_k;\boldsymbol{d}\right); j=1,\cdots,n$ are required either at sample points $\left(0,\cdots,0,v_{k,N}^{(j)}\right)$ to obtain coefficients of $y_{k,N}\left(v_{k,N};\boldsymbol{d}\right)$ or at integration points $\left(0,\cdots,0,v_{k,i}^{(j)},0,\cdots,\beta_k\right); i=1,N-1$ to perform an n-point Gauss–Hermite quadrature for the ith integration in (16), (17), and (18) or (19), (21), (22), (23), (24), (25), (26), (27), (28), and (29). The same procedure is repeated for all univariate component functions, i.e., for all $y_{k,i}\left(v_{k,i};\boldsymbol{d}\right), i=1,\ldots,N$, and for all constraint functions, i.e., for all $y_k\left(v_k;\boldsymbol{d}\right), k=1,\ldots,K$. Therefore, the total cost of the proposed univariate method entails a *maximum* of nNK function evaluations. Note that the above cost is in addition to any function evaluations required for locating the MPP in each constraint.

4 Numerical examples

Two example sets, one involving two sensitivity problems and the other involving four RBDO problems, are presented to illustrate the proposed decomposition method. Constraints associated with both mathematical functions (examples 1 and 3) and structural problems (examples 2–6) were included. Comparisons have been made with FORM/SORM and direct Monte Carlo simulation to evaluate the accuracy and efficiency of the new method. SORM sensitivities in examples 1 and 2 are based on constant principal curvatures with respect to design parameters. In solving RBDO problems (examples 4–6), all approximate methods employed the nested double loop for design and reliability iterations. No single-loop FORM-based methods, although available in the current literature, were included, as the objective was to determine how the accuracy and efficiency of a reliability analysis influence an optimization process. All structural sensitivities were obtained by the finite-difference method involving 1% perturbations. The optimization algorithms selected were sequential quadratic programming in examples 3, 4, and 6, and sequential linear programming in example 5. The sensitivity analysis in the univariate method employed the linear approximation of $y_{k,N}\left(v_{k,N};\boldsymbol{d}\right)$ for all RBDO examples.

In solving all example problems with the univariate method, 5 or 7 sample points were selected with $v_{k,i}^{(j)} = v_{k,i}^{*}-(n-1)/2, v_{k,i}^{*}-(n-3)/2,\cdots,v_{k,i}^{*},\cdots,v_{k,i}^{*}+(n-3)/2, v_{k,i}^{*}+(n-1)/2,$ where $v_{k,i}^{*}=0; i=1,N-1$ and $v_{k,N}^{*}=\beta_k$. Following Lagrange interpolations of all univariate component functions, the same value of n was selected as the number of integration points in evaluating various univariate integrals. Hence, the total number of function evaluations required by the univariate method, in addition to those required for locating the MPP, is $(n-1)NK$. When comparing computational efforts by various RBDO methods, the number of *original* performance function evaluations was chosen as the primary metric in this work.

Table 1 Gradients of two mathematical constraint functions

Gradients	Methods			
	FORM	SORM	Univariate (linear)	Monte Carlo[a]
$\partial c_1(\boldsymbol{d})/\partial d_1$	0.007896	0.007896	0.01390	0.01389
$\partial c_1(\boldsymbol{d})/\partial d_2$	−0.007896	−0.007896	−0.00475	−0.00482
$\partial c_2(\boldsymbol{d})/\partial d_1$	0.004132	0.004132	0.001945	0.001944
$\partial c_2(\boldsymbol{d})/\partial d_2$	−0.004132	−0.004132	−0.001945	−0.001944

[a] Sample size=10^6 for each simulation; finite difference with 1% perturbation

4.1 Example set I—design sensitivity analysis

4.1.1 Example 1—elementary mathematical functions

Consider two constraint functions $c_k(\boldsymbol{d}) = P[g_k(\boldsymbol{X};\boldsymbol{d}) < 0]; k=1,2,$ where the cubic and quartic performance functions, respectively, are expressed by

$$g_1(\boldsymbol{X};\boldsymbol{d}) = 2.2257 - \frac{0.025\sqrt{2}}{27}\left(X_1(d_1)+X_2(d_2)-20\right)^3 + \frac{33}{140}\left(X_1(d_1)-X_2(d_2)\right) \tag{30}$$

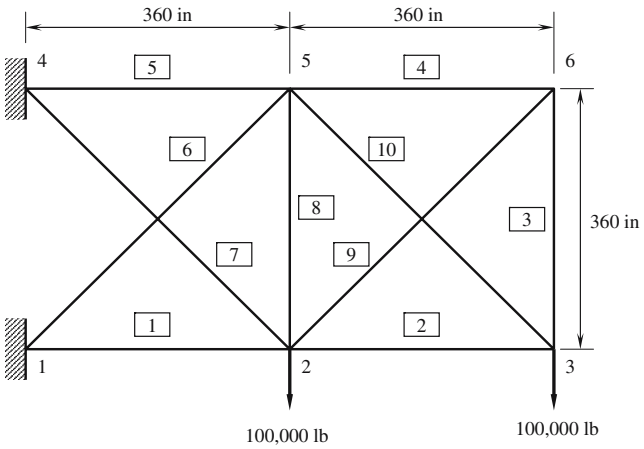

Fig. 3 A ten-bar truss structure; a *boxed number* indicates a member number

Table 2 Gradients of constraint for the ten-bar truss

Gradients[a]	Methods				
	FORM	SORM	Univariate (linear)	Univariate (quadratic)	Monte Carlo[b]
$\partial c(\boldsymbol{d})/\partial d_1$	−0.2107	−0.3156	−0.3038	−0.2851	−0.2824
$\partial c(\boldsymbol{d})/\partial d_2$	−0.0155	−0.0232	−0.0217	−0.0210	−0.0260
$\partial c(\boldsymbol{d})/\partial d_3$	−0.0086	−0.0129	−0.0121	−0.0116	−0.0136
$\partial c(\boldsymbol{d})/\partial d_4$	−0.0086	−0.0129	−0.0121	−0.0116	−0.0124
$\partial c(\boldsymbol{d})/\partial d_5$	−0.2005	−0.3003	−0.2850	−0.2712	−0.2720
$\partial c(\boldsymbol{d})/\partial d_6$	−0.0729	−0.1092	−0.1042	−0.0982	−0.1092
$\partial c(\boldsymbol{d})/\partial d_7$	−0.0769	−0.1152	−0.1094	−0.1037	−0.1144
$\partial c(\boldsymbol{d})/\partial d_8$	0.0007	0.0010	0.0010	0.0009	0.0008
$\partial c(\boldsymbol{d})/\partial d_9$	−0.0250	−0.0375	−0.0357	−0.0338	−0.0352
$\partial c(\boldsymbol{d})/\partial d_{10}$	−0.0464	−0.0695	−0.0664	−0.0626	−0.0640

[a] Truss members 1–10 are identified in Fig. 3.
[b] Sample size=10^6 for each simulation; finite difference with 1% perturbation

and

$$g_2(\boldsymbol{X};\boldsymbol{d}) = \frac{5}{2} + \frac{1}{216}(X_1(d_1) + X_2(d_1) - 20)^4 - \frac{33}{140}(X_1(d_1) - X_2(d_2)) \quad (31)$$

where $\boldsymbol{X}(\boldsymbol{d}) = \{X_1(d_1), X_2(d_2)\}^T$ is a bivariate, independent Gaussian random vector with means μ_i=10 and standard deviations σ_i=3; i=1,2. From an MPP search, $v_1^* = \{0, 2.2257\}^T$ and $\beta_1 = \|v_1^*\| = 2.2257$ for the cubic function, and $v_2^* = \{0, 2.5\}^T$ and $\beta_2 = \|v_2^*\| = 2.5$ for the quartic function. For the univariate method, a value of n=5 was selected, resulting in an additional 16 function evaluations for both functions. The design vector is $\boldsymbol{d} = \{\mu_1, \mu_2\}^T = \{10, 10\}^T$.

Table 1 presents partial derivatives $\partial c_1(\boldsymbol{d})/\partial d_i; i = 1, 2$ and $\partial c_2(\boldsymbol{d})/\partial d_i; i = 1, 2$, calculated by FORM/SORM, the proposed univariate decomposition method (linear approximation), and Monte Carlo simulation using 10^6 samples. The univariate method yields very accurate estimates of gradients of both constraints with a maximum error of less than 1% when compared with simulation results. In contrast, existing FORM/SORM for this particular example contains maximum errors of 64 and 113% for cubic and quartic performance functions, respectively. The SORM results are the same as the FORM results, indicating that there is no improvement over FORM for problems involving inflection point (cubic function) or high nonlinearity (quartic function).

4.1.2 Example 2—ten-bar truss

A linear-elastic, ten-bar truss structure shown in Fig. 3 was studied to examine the accuracy and efficiency of the proposed univariate method for calculating gradients. The Young's modulus of the material is 10^7 psi. Two concentrated forces of 10^5 lb are applied at nodes 2 and 3. The cross-sectional area $X_i(\boldsymbol{d})$ for each bar is independent, follows normal distribution, and has a mean μ_i=2.5 in.^{2} and standard deviation σ_i=0.5 in.^{2}; $i = 1, \cdots, 10$. According to the loading condition, the maximum displacement $[(v_3(X_1(d_1), \cdots, X_{10}(d_{10}))]$ occurs at node 3, where a permissible displacement is limited to 18 in. Therefore, the constraint function is described by $c(\boldsymbol{d}) = P[18 - v_3(X_1(d_1), \cdots, X_{10}(d_{10})) < 0]$.

From an MPP search, the reliability index is $\beta = \|v^*\| = 1.3642$. Table 2 lists ten gradients of the failure probability of the truss, i.e., $\partial c(\boldsymbol{d})/\partial d_i; i = 1, \cdots, 10$, which were calculated using the proposed univariate method employing both linear and quadratic approximations of the Nth univariate function, FORM, SORM, and direct Monte Carlo simulation (10^6 samples). For the

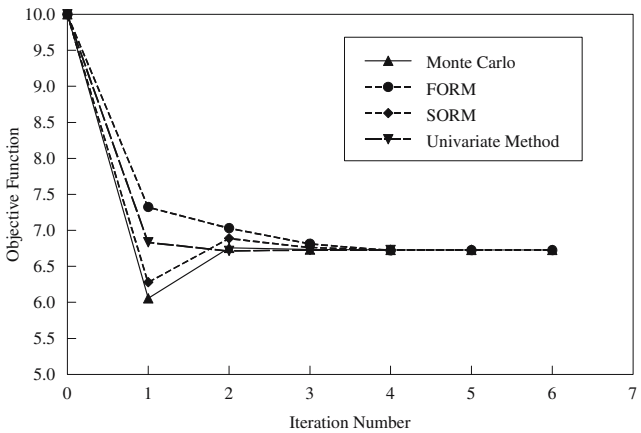

Fig. 4 History of the mathematical objective function

Table 3 Optimization results by various methods for mathematical functions

	Methods[a]			
	FORM	SORM	Univariate	Monte Carlo
No. of iterations	6	5	4	4
No. of function evaluations	1,406	1,226	949	26×10^6
Final design, $\boldsymbol{d}^* = \{d_1^*, d_2^*\}^T$				
d_1^*	3.4391	3.4544	3.4544	3.4547
d_2^*	3.2867	3.2760	3.2740	3.2741
Constraint functions				
$c_1(\boldsymbol{d}^*)-\Phi(-3)$	1.35×10^{-4}	-9.00×10^{-6}	4.00×10^{-6}	0
$c_2(\boldsymbol{d}^*)-\Phi(-3)$	-2.33×10^{-4}	-2.80×10^{-5}	-2.90×10^{-5}	-5.00×10^{-6}
$c_3(\boldsymbol{d}^*)-\Phi(-4)$	-1.17×10^{-5}	-5.67×10^{-5}	-5.67×10^{-6}	-5.67×10^{-6}
Objective function				
$c_0(\boldsymbol{d}^*)$	6.7258	6.7304	6.7284	6.7288

[a] Initial design, $\boldsymbol{d}_0=\{5, 5\}^T$

univariate method, a value of $n=7$ was selected. As can be seen from Table 2, both SORM and the univariate method predict derivatives of the failure probability more accurately than FORM. This is because the univariate method (linear or quadratic) and SORM are able to approximate the performance function embedded in the constraint more accurately than FORM. The total numbers of function evaluations required by the univariate method (linear or quadratic), FORM, SORM, and Monte Carlo simulation are 187, 127, 365, and 10^6, respectively. Therefore, the univariate method is not only accurate, but also more computationally efficient than SORM.

From Table 2, the linear and quadratic options of the univariate method have similar accuracy, although the latter option provides slightly improved results for the two largest gradients ($\partial c(\boldsymbol{d})/\partial d_1$ and $\partial c(\boldsymbol{d})/\partial d_5$). Nonetheless, both options lead to superior gradients when compared with FORM. For simplicity, the linear option was selected for solving the following RBDO problems with the univariate method.

4.2 Example set II—reliability-based design optimization

4.2.1 Example 3—mathematical functions

Consider a mathematical example with two independent Gaussian random variables and three nonlinear constraints. The RBDO problem is defined by

$$\begin{aligned}
\min_{\boldsymbol{d}\in\mathbb{R}^2} \quad & c_0(\boldsymbol{d}) = d_1 + d_2 \\
\text{subject to} \quad & c_1(\boldsymbol{d}) = P\left[\frac{X_1^2(d_1)X_2(d_2)}{20} - 1 < 0\right] \le \Phi(-3) \\
& c_2(\boldsymbol{d}) = P\left[\frac{(X_1(d_1)+X_2(d_2)-5)^2}{30} + \frac{(X_1(d_1)-X_2(d_2)-12)^2}{120} - 1 < 0\right] \le \Phi(-3) \\
& c_3(\boldsymbol{d}) = P\left[\frac{80}{\left(X_1^{5/2}(d_1)+8X_2(d_2)+5\right)} - 1 < 0\right] \le \Phi(-4) \\
& 0 \le d_i \le 10; i = 1, 2
\end{aligned} \tag{32}$$

where $\boldsymbol{X}(\boldsymbol{d}) = \{X_1(d_1), X_2(d_2)\}^T \in \mathbb{R}^2$ is an independent, bivariate Gaussian random vector with means μ_i, and standard deviations $\sigma_i=0.3$; $i=1, 2$. The design vector is $\boldsymbol{d} = \{d_1, d_2\}^T = \{\mu_1, \mu_2\}^T$.

Using the initial design $\boldsymbol{d_0} = \{5, 5\}^T$, Fig. 4 depicts the optimization history when the constraints are evaluated by the univariate decomposition method, FORM, SORM, and Monte Carlo simulation involving 10^6 samples for each failure probability calculation. The detailed results presented in Table 3 suggest that all four methods are able to reach an optimum state in 4–6 iterations, which yield very close optimal solutions. Hence, each method can be used to solve this optimization problem. It is interesting to note that SORM requires fewer function evaluations than FORM, which is somewhat counterintuitive because reliability analysis by SORM is generally more expensive than that by FORM. However, an exception may occur when SORM leads to fewer design iterations than FORM in the outer loop, as observed in this particular problem. Nevertheless, the univariate method is more efficient than FORM or SORM because the fewest number of function evaluations were required to solve this example.

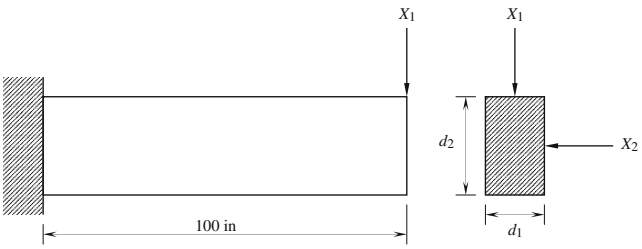

Fig. 5 A cantilever beam subjected to end loads

Table 4 Statistical properties of random input for the cantilever beam

Random variable	Mean	Standard deviation	Probability distribution
X_1, lb	1,000	100	Gaussian
X_2, lb	500	100	Gaussian
X_3, psi	40,000	200	Gaussian
X_4, psi	29×10^6	1.45×10^6	Gaussian

4.2.2 Example 4—cantilever beam

In this example, the design of a fixed cantilever beam with a deterministic length L=100 in., a random vertical load X_1, and a random lateral load X_2, as shown in Fig. 5, was studied. The beam material has random uniaxial yield strength X_3 and random elastic modulus X_4. The objective is to minimize the area of the beam cross-section, where width d_1 and height d_2 are two design variables. Two nonlinear failure modes were examined. The first failure mode is due to yielding at the fixed end of the cantilever, and the second failure mode is associated with the tip displacement exceeding a permissible value of 2.5 in. The RBDO problem is stated as

$$\begin{aligned} \min_{d\in\mathbb{R}^2} \quad & c_0(\boldsymbol{d}) = d_1 d_2 \\ \text{subject to} \quad & c_1(\boldsymbol{d}) = P\left[X_3 - \tfrac{600}{d_1 d_2}\left(\tfrac{X_1}{d_2} + \tfrac{X_2}{d_1}\right) < 0\right] \le \Phi(-2.5) \\ & c_2(\boldsymbol{d}) = P\left[2.5 - \tfrac{4\times10^6}{X_4 d_1 d_2}\sqrt{\tfrac{X_1^2}{d_2^4} + \tfrac{X_2^2}{d_1^4}} < 0\right] \le \Phi(-3.5), \\ & 0 \le d_i \le 5 \text{ inches}; i = 1,2 \end{aligned} \tag{33}$$

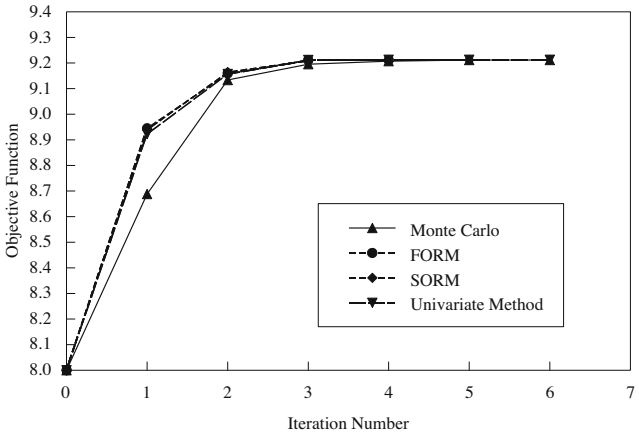

Fig. 6 History of objective function for the cantilever beam

where $\boldsymbol{X} = \{X_1, X_2, X_3, X_4\}^T \in \mathbb{R}^4$ is an independent, four-dimensional Gaussian random vector in which each random variable has the mean and standard deviation listed in Table 4. The design vector is $\boldsymbol{d} = \{d_1, d_2\}^T$. The initial design vector is $\boldsymbol{d}_0 = \{2, 4\}^T$ in.

Figure 6 illustrates the optimization history of the proposed univariate method, FORM, SORM, and Monte Carlo simulation. Table 5 compares the accuracy and efficiency of three approximate methods by using the Monte Carlo benchmark solution. The results suggest that all three methods attain the same optimum value (≈9.21 in.^{2}) of the objective function. The univariate method is slightly more expensive than FORM because of (1) the additional function evaluations required after locating MPPs and (2) the larger design iterations involved in this particular example. Even if the numbers of design iterations are the same, the univariate method will require slightly more

Table 5 Optimization results by various methods for the cantilever beam

	Methods[a]			
	FORM	SORM	Univariate	Monte Carlo
No. of iterations	4	5	6	6
No. of function evaluations	992	1412	1373	29×10^6
Final design, $\boldsymbol{d}^* = \{d_1^*, d_2^*\}^T$				
d_1^*, in.	2.4530	2.4580	2.4683	2.4629
d_2^*, in.	3.7550	3.7476	3.7326	3.7403
Constraint function				
$c_1(\boldsymbol{d}^*) - \Phi(-2.5)$	0	-9.00×10^{-6}	-2.80×10^{-5}	-5.00×10^{-6}
$c_2(\boldsymbol{d}^*) - \Phi(-3.5)$	4.24×10^{-5}	2.14×10^{-5}	-2.26×10^{-5}	-2.40×10^{-6}
Objective function				
$c_0(\boldsymbol{d}^*)$, in.^{2}	9.2109	9.2117	9.2132	9.2119

[a] Initial design, $\boldsymbol{d}_0 = \{2, 4\}^T$ in.

Table 6 Failure probability of the cantilever beam at optimum design

Methods	$c_1(\boldsymbol{d}^*)$[a]	$c_2(\boldsymbol{d}^*)$[b]	Comment
FORM	0.006210	0.0002750	Violates 2nd constraint by 18%
SORM	0.006201	0.0002540	Violates 2nd constraint by 9%
Univariate	0.006182	0.0002100	Satisfies both constraints
Monte Carlo	0.006205	0.0002350	Satisfies both constraints

[a] The allowable value for the first constraint is $\Phi(-2.5)=0.00620968$.
[b] The allowable value for the second constraint is $\Phi(-3.5)=0.000232673$.

function evaluations than FORM. In this example, both FORM and univariate methods are more efficient than SORM, a trend that is expected unless the number of design iterations required by SORM is significantly fewer than others.

Because the univariate method and FORM/SORM entail approximate reliability analysis, the constraints associated with the optimal design generated by each method were evaluated using the Monte Carlo simulation (10^6 samples). Table 6 presents the values of the failure probability embedded in each constraint. It appears that both FORM and SORM slightly violate the second constraint with a maximum error of 18 and 9%, respectively, in calculating the failure probability. In contrast, no such violations are observed in the univariate method. This is because the proposed univariate method is more accurate than FORM/SORM in performing reliability analysis in this example.

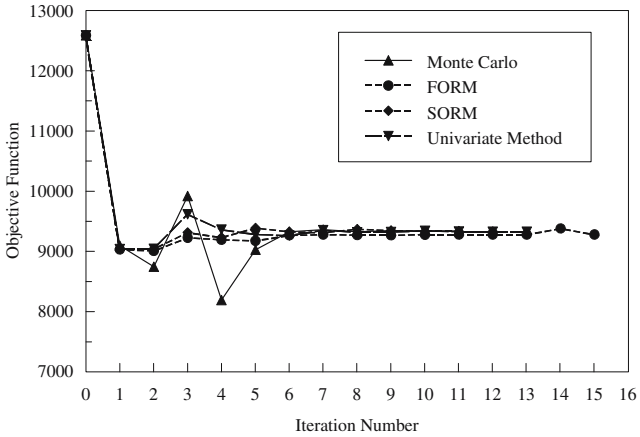

Fig. 7 History of objective function for the ten-bar truss

4.2.3 Example 5—ten-bar truss

A ten-bar truss, illustrated in Fig. 3, was designed by minimizing its total volume given that the truss reliability is no less than a target value of $\Phi(2) = 0.9772$. The RBDO formulation is

$$\begin{aligned} \min_{\boldsymbol{d}\in\mathbb{R}^{10}} \quad & c_0(\boldsymbol{d}) = 360\left[d_1 + d_2 + d_3 + d_4 + d_5 + d_8 + \sqrt{2}(d_6 + d_7 + d_9 + d_{10})\right] \\ \text{subject to} \quad & c_1(\boldsymbol{d}) = P[14 - v_3(X_1(d_1), \cdots, X_{10}(d_{10})) < 0] \le \Phi(-2) \\ & 0 \le d_i \le 5 \; in.; i = 1, 10 \end{aligned} \tag{34}$$

where $\boldsymbol{X} = \{X_1(d_1), \cdots, X_{10}(d_{10})\}^T \in \mathbb{R}^{10}$ is an independent Gaussian random vector, with each component representing a random cross-section of the truss. The random variable

Table 7 Optimization results by various methods for the ten-bar truss

	Methods[a]			
	FORM	SORM	Univariate	Monte Carlo
No. of iterations	15	12	13	11
No. of function evaluations	2,694	4,894	3,113	117×10^5
Final design, $\boldsymbol{d}^* = \{d_1^*, \ldots, d_{10}^*\}^T$ (in.^{2})				
d_1^*	3.998	4.072	3.935	4.21
d_2^*	1.944	1.968	1.947	1.918
d_3^*	1	1	1	1
d_4^*	1	1	1	1
d_5^*	4.388	4.384	4.381	4.268
d_6^*	2.827	2.783	2.94	2.669
d_7^*	2.225	2.284	2.117	2.436
d_8^*	1	1.001	1	1.001
d_9^*	1	1	1	1.039
d_{10}^*	2.754	2.77	2.743	2.728
Constraint function				
$c_1(\boldsymbol{d}^*)-\Phi(-2)$	1.25×10^{-2}	-1.13×10^{-3}	2.3×10^{-4}	2.3×10^{-5}
Objective function (in.^{3})				
$c_0(\boldsymbol{d}^*)$	9,282	9,332	9,327	9,340

[a] Initial design, $\boldsymbol{d}_0=\{3, ..., 3\}^T$ in.^{2}

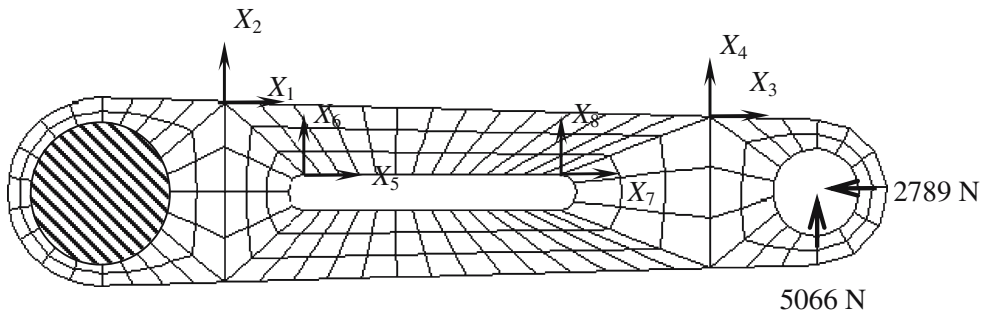

Fig. 8 Initial design of the torque-arm geometry at mean values of shape parameters

X_i follows Gaussian distribution and has means μ_i and standard deviation σ_i=0.2 in.^{2}; $i = 1, \cdots, 10$. The design vector is $\boldsymbol{d} = \{d_1, \cdots, d_{10}\}^T = \{\mu_1, \cdots, \mu_{10}\}^T$. The initial design point is $\boldsymbol{d}_0 = \{3, \cdots, 3\}^T$ in.^{2}.

Figure 7 and Table 7 present the optimization history and optimization results by various methods, respectively. The optimal volumes achieved by the univariate method, SORM, and Monte Carlo vary from 9,327 to 9,340 in.^{3}. In contrast, FORM leads to a lower optimal volume, which is 9,282 in.^{3}. A Monte Carlo reliability analysis at optimal designs obtained by FORM, SORM, and the univariate method reveals that the failure probability estimates have associated absolute errors of 55, 5, and 1%, respectively. Hence, FORM violates the constraint leading to the lower optimum volume of the truss. Both SORM and the univariate method satisfy the constraint and, hence, provide acceptable designs. However, the univariate method proposed is more efficient than SORM in solving the truss problem.

4.2.4 Example 6—torque arm

The final example involves designing a torque-arm, where eight random shape parameters $X_i(d_i); i = 1, 8$ describe its outer and inner boundaries, as shown in Fig. 8, for the mean input at the initial design. The left hole of the structure is fixed, and two deterministic forces F_1=2,789 N and F_2=5,066 N are applied at the center of the right hole. The torque-arm material has a mass density ρ=7,800 kg/m^3, elastic modulus E=207 GPa, Poisson's ratio ν= 0.3, and uniaxial yield strength S_y=400 MPa. The objective is to minimize the mass of the structure $m(\boldsymbol{d})$ by changing the shape of the geometry (i.e., by $\boldsymbol{X}(\boldsymbol{d}) \in \mathbb{R}^8$) such that the von Mises stresses at five selected points do not exceed S_y. The locations of these five points, marked as finite element nodes 90, 98, 106, 173, and 175, are illustrated in Fig. 9. Mathematically,

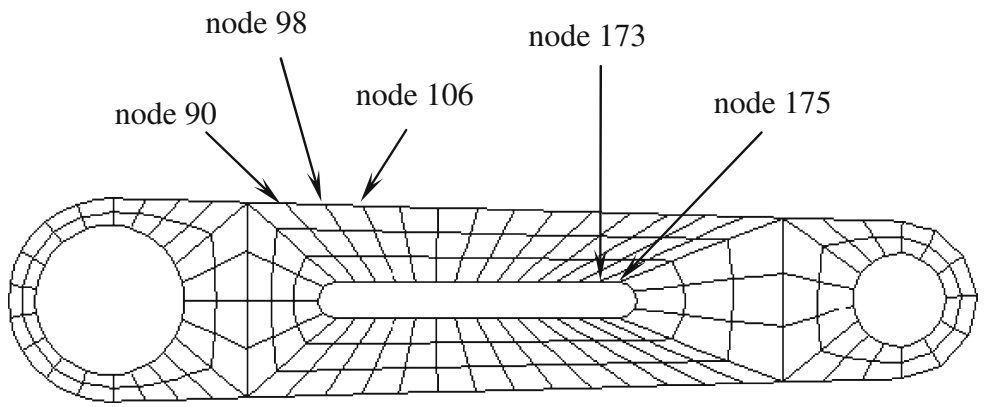

Fig. 9 Locations of points for prescribing constraints of the torque arm

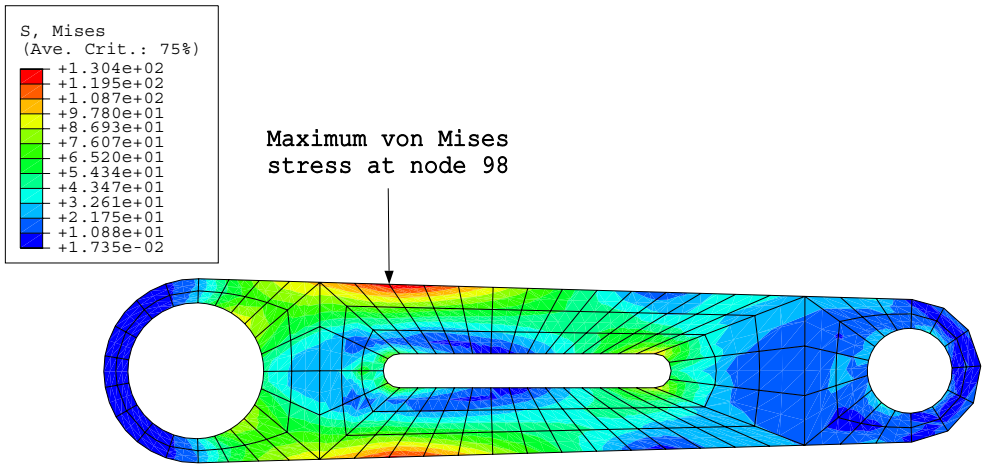

Fig. 10 Contour of von Mises stress at mean values of shape parameters from initial design

$$
\begin{aligned}
\min_{\boldsymbol{d}\in\mathbb{R}^{10}} \quad & c_0(\boldsymbol{d}) = m(\boldsymbol{d}) \\
\text{subject to} \quad & c_k(\boldsymbol{d}) = P\left[S_y - \sigma_{k,e}(\boldsymbol{X};\boldsymbol{d}) < 0\right] \le \Phi(-3); k = 1,5 \\
& -1\text{mm} \le d_1 \le 1\text{mm} \\
& -1\text{mm} \le d_2 \le 1\text{mm} \\
& -1\text{mm} \le d_3 \le 1\text{mm} \\
& -2\text{mm} \le d_4 \le 1\text{mm} \\
& -5\text{mm} \le d_5 \le 1\text{mm} \\
& -0.5\text{mm} \le d_6 \le 2\text{mm} \\
& -1\text{mm} \le d_7 \le 6\text{mm} \\
& -0.5\text{mm} \le d_8 \le 1\text{mm}
\end{aligned}
\tag{35}
$$

where $\sigma_{k,e}(\boldsymbol{X};\boldsymbol{d})$ is the von Mises equivalent stress at the kth selected point. The finite element mesh includes 657 nodes and 177 eight-noded quadrilateral elements. A plane stress condition was assumed. The independent random vector $\boldsymbol{X}$, which represents manufacturing variability, follows Gaussian distribution. The components X_i have means μ_i and standard deviations σ_i=0.2 mm; $i = 1, \cdots, 10$. The design vector is $\boldsymbol{d} = \{d_1, \cdots, d_8\}^T = \{\mu_1, \cdots, \mu_8\}^T$.

The initial design point is $\mathrm{d}_0 = \{0, \cdots, 0\}^T$ *mm* with the corresponding finite element mesh depicted in Fig. 9. Following linear-elastic stress analysis, Fig. 10 presents the contour plot of the von Mises stress at the initial design when shape parameters assume their mean values. Due to conservative initial design, the maximum von Mises stress of 130 MPa, which occurs at node 98, is much lower than the uniaxial yield strength (S_y=400 MPa). During design iterations, the nodal movements that control shape parameters $X_i(d_i); i = 1, 8$ were performed by a design velocity field involving isoparametric mapping (Choi and Chang 1994).

For computational efficiency, the optimal design was obtained in two steps. In the first step, a coarse RBDO was

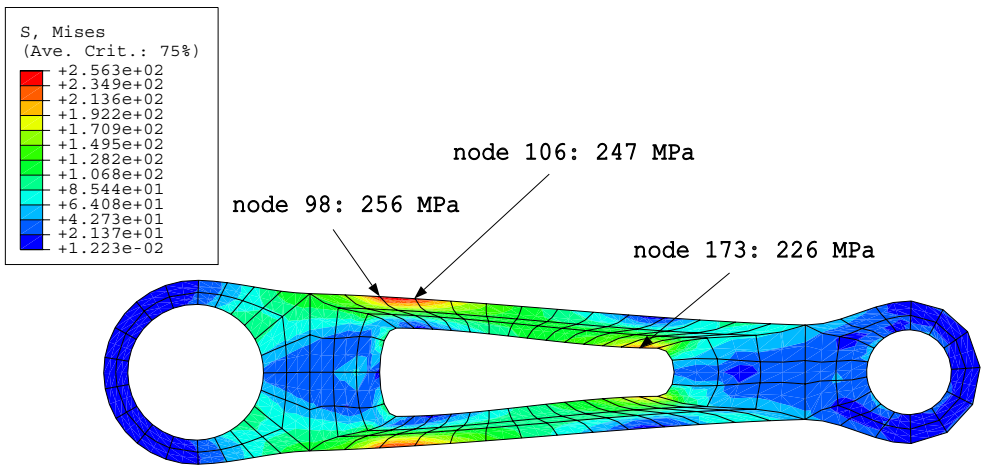

Fig. 11 Contour of von Mises stress at mean values of shape parameters from reliability-based optimum design

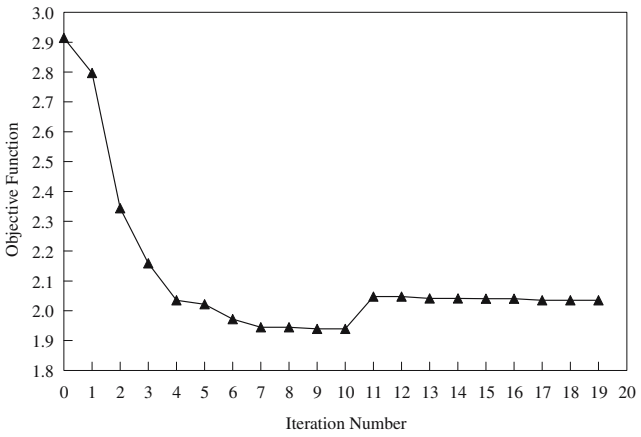

Fig. 12 Optimization history of objective function for the torque arm

performed using the initial design $\mathrm{d}_0 = \{0, \cdots, 0\}^T$ and an approximate reliability method, known as the mean-value first-order second moment method (Madsen et al. 1986). The resultant design after ten iterations in the first step (coarse RBDO) is $\overline{\boldsymbol{d}}_0 = \{-0.427, -1, -0.063, -2, -0.327, 2, 0.234, 0.875\}^T$mm. In the second step, a refined RBDO involving the proposed univariate method and the result of step 1 as the initial design (i.e., $\boldsymbol{d}_0 = \overline{\boldsymbol{d}}_0$) was employed. After nine iterations, the final design was attained, which is $\boldsymbol{d}^* = \{-0.709, -0.721, -0.077, -2, -0.247, 2, 0.258, 0.524\}^T$mm, with the corresponding mean shape presented in Fig. 11. The optimal mass of the torque arm is 2.035 kg—a 30% reduction from the initial mass of 2.915 kg. Figure 11 also displays the contour plot of the von Mises stress at the optimal design when the shape parameters assume their mean values. Compared with the conservative initial design of Fig. 10, larger stresses, for example, 256, 247, and 226 MPa at nodes 98, 106, and 173, respectively, can be safely tolerated in the final design of Fig. 11. The larger area of the slotted hole and movement of outer boundaries have led to significant alteration of the shape of the initial design. Figure 12 shows the optimization history of the objective function.

If the uncertainty of $\boldsymbol{X}$ is ignored and the constraints in (35) are replaced by $S_y - \sigma_{k,e}(\boldsymbol{d}; \boldsymbol{d})0; k = 1, 5$, as commonly adopted in traditional design optimization, 13 iterations led to $\boldsymbol{d}^* = \{-1, -1, -0.062, -2, -1.785, 2, 1.676, 0.795\}^T$mm and a corresponding optimal mass of 1.861 kg—a 36% reduction from the initial mass. Therefore, a traditional risk-ignoring optimization process may lead to a smaller mass than that obtained from RBDO, but with higher stresses, as depicted in the contour plot of Fig. 13. If uncertainties are included, the optimal design in Fig. 13 is highly likely to violate the reliability constraints. By comparing optimal designs from RBDO (Fig. 11) and risk-ignoring optimization (Fig. 13), it appears that the outer boundaries generated by both designs are similar. However, the inner slot from the RBDO is smaller than that from the risk-ignoring optimization. The primary reason is that the latter optimization does not account for variability of shape parameters and of the performance function. In addition, the sensitivity of the von Mises stress with respect to shape parameters in the inner boundary is much larger than that in the outer boundary. Finally, a few finite elements in both meshes of Figs. 11 and 13 are distorted. Although remeshing or mesh-free computations during design iterations should alleviate this problem, the fundamental observations discussed above will remain unaltered. No remeshing was performed in the present work.

In summary, the univariate method consistently provides very accurate RBDO solutions. Of the three methods

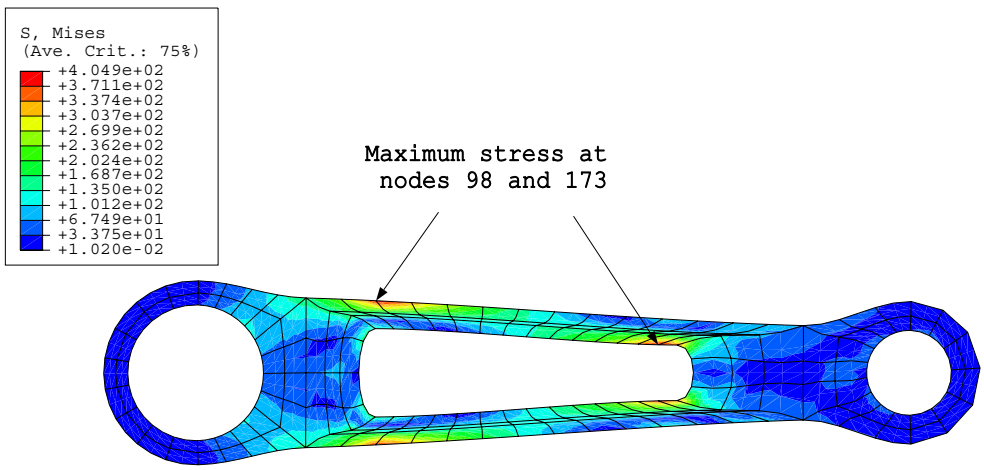

Fig. 13 Contour of von Mises stress at mean values of shape parameters from risk-ignoring optimum design

studied, the FORM-based RBDO is the most efficient method; however, it may lead to infeasible or inaccurate designs. Both SORM and the univariate method have comparable accuracies, but the univariate method is less expensive than SORM. Nevertheless, for industrial-scale design applications, further research is required in making the proposed univariate method computationally more efficient by potentially decoupling the design and reliability iterations or exploring the possibility of single-loop formulations.

5 Conclusions and outlook

A new univariate decomposition method was developed for design sensitivity analysis and RBDO of mechanical systems subject to uncertain performance functions in constraints. The method involves a novel univariate approximation of a general multivariate function in the rotated Gaussian space for reliability analysis, analytical sensitivity of failure probability with respect to design variables, and standard gradient-based optimization algorithms. In both reliability and design sensitivity analyses, the proposed effort can be viewed as performing multiple one-dimensional integrations. The evaluation of these one-dimensional integrations requires calculating only conditional responses at selected deterministic input determined by sample points and Gauss–Hermite integration points. More importantly, the design sensitivities are calculated without incurring additional function evaluations to those required for reliability analysis. Results of two numerical examples involving mathematical functions and truss problems indicate that the proposed method provides reasonably accurate and computationally efficient estimates of the sensitivity of failure probability. Subsequent results of four design problems, entailing mathematical functions and structural/solid-mechanics applications, indicate that the new sensitivity equations developed facilitate an accurate and/or efficient method for RBDO of mechanical systems. Although significant strides are made, further research is required in potential decoupling or single-loop formulation in conjunction with the univariate method.

Acknowledgment The authors would like to acknowledge the financial support from the US National Science Foundation under grant no. DMI-0355487.

References

Abramowitz M, Stegun I (1972) Handbook of mathematical functions, 9th (edn). Dover, New York, NY

Agarwal H, Renaud JE (2006) New Decoupled framework for reliability-based design optimization. AIAA J 44:1524–1531

Bjerager P (1988) Probability integration by directional simulation. J Eng Mech 114(8):1285–1302

Chiralaksanakul A, Mahadevan S (2005) First-order approximation methods in reliability-based design optimization. J Mech Des 127:851–857

Choi KK, Chang KH (1994) A study of design velocity field computation for shape optimal design. Finite Elem Anal Des 15:241–317

Der Kiureghian A, Dakessian T (1998) Multiple design points in first and second-order reliability. Struct Saf 20(1):37–49

Ditlevsen O, Madsen HO (1996) Structural reliability methods. Wiley, Chichester

Du X, Chen W (2004) Sequential optimization and reliability assessment method for efficient probabilistic design. J Mech Des 126:225–233

Enevoldsen I, Sorensen JD (1994) Reliability-based optimization in structural engineering. Struct Saf 15:169–196

Kirjner-Neto C, Polak E, Der Kiureghian A (1998) An outer approximation approach to reliability-based optimal design of structures. J Optim Theory Appl 98:1–16

Kuschel N, Rackwitz R (1997) Two basic problems in reliability-based structural optimization. Math Methods Oper Res 46:309–333

Liang J, Mourelatos ZP, Tu J (2004) A single-loop method for reliability-based design optimization. Proceedings of the 30th ASME design automation conference, Salt Lake City, UT

Madsen HO, Krenk S, Lind NC (1986) Methods of structural safety. Prentice-Hall, Englewood Cliffs, NJ

Nie J, Ellingwood BR (2000) Directional methods for structural reliability analysis. Struct Saf 22:233–249

Qu X, Haftka RT (2004) Reliability-based design optimization using probabilistic sufficiency factor. Struct Multidiscipl Optim 27:314–325

Rackwitz R (2001) Reliability analysis—a review and some perspectives. Struct Saf 23(4):365–395

Rahman S, Wei D (2006) A univariate approximation at most probable point for higher-order reliability analysis. Int J Solids Struct 43:2820–2839

Royset J, Polak E (2004) Reliability-based optimal design using sample average approximations. Probab Eng Mech 19:331–343

Tu J, Choi K, Park Y (1999) A new study on reliability-based design optimization. ASME J Mech Des 121:557–564

Wei D, Rahman S (2007) Structural reliability analysis by univariate decomposition and numerical integration. Probab Eng Mech 22:27–38

Xu H, Rahman S (2005) Decomposition methods for structural reliability analysis. Probab Eng Mech 20:239–250

Zou T, Mahadevan S (2006) A direct decoupling approach for efficient reliability-based design optimization. Struct Multidiscipl Optim 31:190–200