



Design sensitivity analysis with polynomial chaos for robust optimization

Chengkun Ren¹ · Fenfen Xiong¹ · Bo Mo¹ · Anik Chawdhury¹ · Fenggang Wang¹

Received: 22 May 2020 / Revised: 23 June 2020 / Accepted: 24 July 2020 / Published online: 10 September 2020
© Springer-Verlag GmbH Germany, part of Springer Nature 2020

Abstract

Polynomial chaos (PC) methods with Gauss-type quadrature formulae have been widely applied for robust design optimization. During the robust optimization, gradient-based optimization algorithms are commonly employed, where the sensitivities of the mean and variance of the output response with respect to design variables are calculated. For robust optimization with computationally expensive response functions, although the PC method can significantly reduce the computational cost, the direct application of the classical finite difference method for the analysis of the design sensitivity is impractical with a limited computational budget. Therefore, in this paper, a semi-analytical design sensitivity analysis method based on the PC method is proposed, in which the sensitivity is directly derived based on the Gauss-type quadrature formula without additional function evaluations. Comparative studies conducted on several mathematical examples and an aerodynamic robust optimization problem revealed that the proposed method can reduce the computational cost of robust optimization to a certain extent with comparable accuracy compared with the finite difference-based PC method.

Keywords Robust design optimization · Polynomial chaos · Gauss-type quadrature · Design sensitivity

Nomenclature

b	Coefficients of polynomial chaos model	q	Random parameter variable
d	Dimension of design variables	ω	Gauss-type quadrature weight coefficient
N	Number of function evaluations	$y(x)$	Output response
m	Dimension of random parameters	μ	Mean value
$E(\bullet)$	Operation of calculation expectation	σ	Standard deviation value
$f(x)$	Objective function	Φ	Multi-dimensional orthogonal polynomial
$g(x)$	Constraint function	φ	One-dimensional orthogonal polynomial basis function
p	Order of polynomial chaos model	DSA	Design sensitivity analysis
K	Accuracy level of sparse grid	FDM	Finite difference method
\mathbf{x}	Random design vector	FFNI	Full factorial numerical integration
x	Random design variable	MCS	Monte Carlo simulation
\mathbf{q}	Random parameter vector	SGNI	Sparse grid numerical integration

Responsible Editor: Erdem Acar

Electronic supplementary material The online version of this article (<https://doi.org/10.1007/s00158-020-02704-2>) contains supplementary material, which is available to authorized users.

✉ Fenfen Xiong
fenfenx@bit.edu.cn

¹ School of Aerospace Engineering, Beijing Institute of Technology, Beijing 100081, China

1 Introduction

The objective of robust design optimization is the optimization of the product performance while minimizing the sensitivity of the performance to uncertainties, which has been widely applied to aerospace engineering (Xiong et al. 2015a; Li et al. 2013; Wu et al. 2018), automobile engineering (Zhang et al. 2007), and civil engineering (Cheng et al. 2016) fields. As is common knowledge, a key component of

robust design optimization is the estimation of the mean and variance of the output response, which can be achieved by the use of a metamodel to replace the original output response (Rehman et al. 2014; Janusevskis and Rodolphe 2013). An alternative method involves the development of efficient uncertainty propagation approaches, among which the polynomial chaos (PC) method has been widely studied and applied, as it provides a solid mathematical foundation and fast convergence rate (Xiu and Karniadakis 2002). With the PC method, a random variable can be represented as a stochastic metamodel, and the mean and variance of the random variable can then be analytically obtained. Wu et al. (Wu et al. 2017) and Dodson and Parks (Dodson and Parks 2015) respectively applied the PC method to aerodynamic robust optimization design. Suryawanshi and Ghosh employed the PC method to analyze the aeroelastic stability problems on a rectangular unswept cantilever wing model (Suryawanshi and Ghosh 2016). Xia et al. applied an adaptive PC method to quantify the performance impact of flow and geometric uncertainties for a turbine blade (Xia et al. 2019). Fisher et al., Xiong et al., and Wang et al. respectively applied the PC method to robust trajectory optimization (Fisher and Bhattacharya 2011; Xiong et al. 2015b; Wang et al. 2019a), to increase the robustness of the planned trajectory in response to various disturbances that is beneficial to closed-loop control. In addition to the aleatory uncertainty, the PC method was proposed for the realization of robust design optimization under mixed (aleatory and epistemic) uncertainties, in which the evaluation method for the robustness index under mixed uncertainties was presented (Zhang and Serhat 2013; Zhang 2013). The focus of this study was on PC-based robust optimization under the consideration of aleatory uncertainties.

It is common knowledge that during the gradient-based robust optimization process, the calculation of the design sensitivity is required, i.e., the sensitivities of the first two statistical moments (mean and standard deviation) of the output response with respect to the design variables. The commonly used method for sensitivity analysis is the finite difference method (FDM). For robust optimization, although the PC method can significantly reduce the computational cost when compared with the traditional Monte Carlo simulation (MCS) method for statistical moment calculations, a certain number of additional function evaluations are required for the design sensitivity analysis when employing the traditional FDM. In addition, there is a significant increase in the number of function evaluations required for high-dimensional problems. However, for practical engineering design problems, nonlinear, high-dimensional, and time-consuming high-fidelity simulation is typically required, such as the finite element analysis (FEA) and the computational fluid dynamics (CFD). Furthermore, the direct application of FDM to the analysis of the design sensitivity during robust optimization is impractical with a limited computational budget.

Design sensitivity analysis has been previously investigated in relation to robust design optimization (Keshavarzzadeh et al. 2016; Keshavarzzadeh et al. 2017; Kumar et al. 2018; Miranda et al. 2016; Du and Leifsson 2019; Rumpfkeil 2013) and reliability-based design optimization (Keshavarzzadeh et al. 2016; Keshavarzzadeh et al. 2017; Cho et al. 2016; Lee et al. 2011a; Lee et al. 2011b), for the improvement of the efficiency and accuracy of design. The studies conducted by Cho et al., Lee et al., and Choi et al. in refs. (Cho et al. 2016; Lee et al. 2011a; Lee et al. 2011b) were focused on reliability-based design optimization, in which the MCS method was employed for reliability analysis, and the design sensitivity of the probability of failure was derived using a first-order score function. Rumpfkeil developed a gradient-based robust optimization algorithm based on kriging, in which the design sensitivity was derived analytically based on the kriging metamodel (Rumpfkeil 2013). For PC-based design optimization under uncertainties, several studies have been conducted to develop design sensitivity approaches to save computational cost (Keshavarzzadeh et al. 2016; Keshavarzzadeh et al. 2017; Kumar et al. 2018; Miranda et al. 2016; Du and Leifsson 2019). In these works, the PC method was applied for uncertainty quantification and the design sensitivity was derived based on the PC model by using the adjoint method. However, it is limited to systems with known governing equations, and user familiarity with the physical system mechanism is required, given that the sensitivity information from adjoint method is utilized. In practical engineering, oftentimes the response models in optimization design are computationally expensive black-box-type response functions, such as the CFD/FEA simulation model that has been calibrated by specialists, and the complex coupled multidisciplinary car or flight vehicle system involving various simulation models from different disciplines, and it is difficult or even impossible to derive the governing equation of the entire system. Therefore, the abovementioned design sensitivity method for PC-based design optimization under uncertainties in refs. (Keshavarzzadeh et al. 2016; Keshavarzzadeh et al. 2017; Kumar et al. 2018; Miranda et al. 2016; Du and Leifsson 2019) may be difficult to apply in this case.

For the construction of the PC model for the statistical moment estimation, a commonly used method for the calculation of the PC coefficients is the least square estimation method (Ghisu and Shahpar 2017; Ghisu and Shahpar 2018). An alternative method is the Galerkin projection method (Xiu and Karniadakis 2003), in which the PC coefficients are calculated using a series of Gaussian-type quadrature formulae, such as the full factorial numerical integration (Lee et al. 2009; Lee et al. 2008) and sparse grid numerical integration (Xiong et al. 2010). The Gaussian quadrature is formulated as a weighted summation of the function response values on several Gaussian quadrature nodes, which provides a potential method for the calculation of the design sensitivity of

statistical moments. Therefore, in this study, a semi-analytic design sensitivity analysis method was developed for the PC-based robust optimization, for the reduction of the computational cost. The design sensitivities of the first two statistical moments with respect to the design variables are directly derived based on the Gaussian quadrature formula employed for the statistical moment estimation within the PC method by using the chain rule in conjunction with the FDM strategy, instead of referring to several additional function evaluations, as is the case with the direct FDM. Moreover, when compared with the existing approach by Keshavarzzadeh et al., the proposed method can be more conveniently applied to systems with black-box-type response functions (Keshavarzzadeh et al. 2016; Keshavarzzadeh et al. 2017).

The remainder of this paper is organized as follows. In Section 2, the Gaussian quadrature-based PC method is briefly reviewed. In Section 3, the proposed semi-analytic design sensitivity analysis method is described in detail. Section 4 presents comparative studies on the robust optimization of several mathematical examples, in addition to an airfoil aerodynamic optimization problem, for the verification of the effectiveness of the proposed method. The conclusion is then presented in Section 5.

2 Review of Gaussian quadrature-based PC method in robust optimization

In this study, given that the probability-based robust optimization with random uncertainty was considered, a typical and widely applied robust design optimization problem that can be classified as “sensitivity robustness approach” (Beyer and Sendhoff 2007) was addressed, which can be expressed as follows:

$$\begin{aligned}
 & \text{Find } \mu_{x_1}, \dots, \mu_{x_d} \\
 & \text{Min } F(\mathbf{x}) = w_1 \frac{\mu_f(\mathbf{x}, \mathbf{q})}{\mu_f^*} + w_2 \frac{\sigma_f(\mathbf{x}, \mathbf{q})}{\sigma_f^*} \\
 & \text{s.t. } G_i(\mathbf{x}) = \mu_{g_i}(\mathbf{x}, \mathbf{q}) - k^* \sigma_{g_i}(\mathbf{x}, \mathbf{q}) \geq 0 \quad (i = 1, \dots, nc) \\
 & \mu_{x_j}^L < \mu_{x_j} < \mu_{x_j}^U \quad (j = 1, \dots, d)
 \end{aligned} \tag{1}$$

where $\mathbf{x} = [x_1, \dots, x_d]$ is the random design vector with mean values of $\mu_{x_1}, \dots, \mu_{x_d}$; $\mathbf{q} = [q_1, \dots, q_m]$ is the random parameter vector; μ and σ represent the mean and standard deviation of the variable, respectively; the subscripts f and g denote the objective and constraint functions corresponding to the original deterministic optimization problem, respectively; k^* is a constant employed in robust optimization to ensure the robustness of optimal solution that is typically set as $k^* = 3$; μ_f^* and σ_f^* are the constant scaling factors specified by users to scale μ_f and σ_f to the same magnitude that is beneficial to find the optimum; and w_1 w_2 are weight coefficients defined by the users with some preferences or a priori assumptions.

Figure 1 presents the flowchart of the PC-based robust design optimization using the gradient-based optimization

algorithm. During each optimization iteration, the mean and standard deviation values of the original objective $f(\mathbf{x})$ and constraint functions $g_i(\mathbf{x})$ ($\mu_f, \sigma_f, \mu_{g_i}$, and σ_{g_i}) are calculated using the PC method at the current design point. Moreover, the procedure can be briefly expressed as follows, by using the response function $y = g(\mathbf{x})$ for illustration.

Step 1: represent the output response y as a p th-order PC model with respect to the random design variables \mathbf{x} and random parameters \mathbf{q} using the generalized PC method (Xiu and Karniadakis 2003). Here, p represents the maximum order of the polynomial terms ($\Phi_i(\xi)$, $i = 0, \dots, P$) in the PC model.

$$y \approx \sum_{i=0}^P b_i \Phi_i(\xi) \tag{2}$$

where $\xi = [\xi_1, \dots, \xi_d, \xi_{d+1}, \dots, \xi_{d+m}]$ represents the standard random vector transformed from the original random vector $X = [x_1, \dots, x_d, q_1, \dots, q_m]$ based on their distribution information (Xiu and Karniadakis 2002); the number of polynomial terms is $P + 1 = (p + d + m)! / (d + m)! / p! = (p + D)! / D! / p!$ with D as the sum of the dimensions of x and q ; and $\Phi_i(\xi) = \prod_{k=1}^D \varphi_i^k(\xi_k)$ is the product of the one-dimensional orthogonal polynomial basis function ($\varphi_i^k(\xi_k)$) that corresponds to the standard random variable in each dimension.

Step 2: calculate the PC coefficient b_i ($i = 0, \dots, P$).

Using the Galerkin projection method, the two sides of Eq. (2) are projected onto each orthogonal polynomial $\Phi_i(\xi)$. Then, each PC coefficient is calculated as follows using the orthogonality of the orthogonal polynomials.

$$b_i = E[y\Phi_i(\xi)] / E[\Phi_i(\xi)\Phi_i(\xi)], \quad (i = 0, 1, \dots, P) \tag{3}$$

where $E[\cdot]$ indicates the expectation.

The computational cost of Eq. (3) is mainly due to the calculation of the numerator, which can be carried out using a Gaussian quadrature-type numerical integration method such as the commonly used full factorial numerical integration (FFNI) (Lee et al. 2009; Lee et al. 2008) for lower-dimensional problems, or sparse grid numerical integration (SGNI) for higher-dimensional problems (Ghisu and Shahpar 2017). With FFNI, the numerator can be calculated using a direct full tensor product Gaussian quadrature formula, as expressed below.

$$E[y\Phi_i(\xi)] = \sum_{j_1=1}^{m_1} \omega_{1,j_1} \dots \sum_{j_D=1}^{m_D} \omega_{D,j_D} y(x_{1,j_1}, \dots, x_{D,j_D}) \Phi_i(\xi) \tag{4}$$

where $\omega_{j,k}$ and $x_{j,k}$ are the k th weight and one-dimensional node for the j th input random variable of $X = [x_1, \dots, x_d, q_1, \dots, q_m]$, respectively; m_j represents the corresponding number of nodes; and the value of ξ employed in $\Phi_i(\xi)$ is calculated by transforming from the d -dimensional node $(x_{1,j_1}, \dots, x_{D,j_D})$ based on the distribution information of X .

With SGNI, the D -dimensional sparse grid nodes with an accuracy level K are first obtained using a special tensor product operation, as follows:

$$U_D^K = \bigcup_{K+1 \leq |i| \leq Q} U_1^{i_1} \otimes U_1^{i_2} \otimes \dots \otimes U_1^{i_D} \quad (5)$$

where i_1, i_2, \dots, i_D determines the number of nodes in each dimension $m_j (j = 1, \dots, D)$, $|i| = i_1 + i_2 + \dots + i_D$, $Q = K + D$; and U_D^K represents the collection of sparse grids, i.e., the combination of all the multi-dimensional nodes produced by a certain number of direct smaller-size tensor products.

For each combination $\{i_1, \dots, i_D\}$ that satisfies $K + 1 \leq |i| \leq Q$, the direct full tensor product is used to generate D -dimensional nodes. Based on Eq. (5), a total of N D -dimensional nodes $\{\mathbf{x}_1, \dots, \mathbf{x}_k, \dots, \mathbf{x}_N\}$ can be obtained, and the weight coefficient for $\mathbf{x}_k = [x_{1,j_1}^k, \dots, x_{D,j_D}^k]$ is calculated as follows:

$$\omega_k = (-1)^{Q-|i|} \binom{D-1}{Q-|i|} (\omega_{1,j_1} \dots \omega_{D,j_D}) \quad (6)$$

where $\omega_{1,j_1} \dots \omega_{D,j_D}$ denotes the product of the one-dimensional weights.

Thus, $E[y\Phi_i(\xi)]$ can be calculated as follows:

$$E[y\Phi_i(\xi)] = \sum_{k=1}^N \omega_k (y(\mathbf{x}_k) \Phi_i(\xi)) \quad (7)$$

Step 3: when the PC coefficients are obtained, estimate the first two statistical moments of the random output response y .

$$\mu_y = E[y] = \sum_{i=0}^P b_i E[\Phi_i(\xi)] = b_0 \quad (8)$$

$$\sigma_y^2 = E[y^2] - E[y]^2 = E \left[\left(\sum_{i=0}^P b_i \Phi_i(\xi) \right)^2 \right] - \left(\sum_{i=0}^P b_i E[\Phi_i(\xi)] \right)^2 = \sum_{i=1}^P b_i^2 E[\Phi_i^2(\xi)] \quad (9)$$

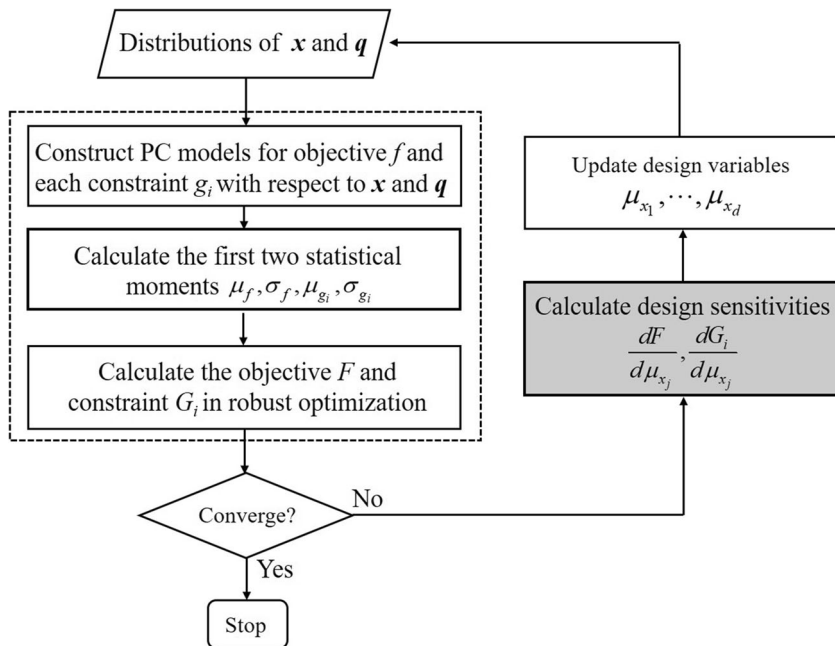
3 Proposed semi-analytical design sensitivity analysis method based on PC

As shown in Fig. 1, in addition to the statistical moment calculation, the design sensitivity $(\frac{dF}{d\mu_{x_j}}, \frac{dG_i}{d\mu_{x_j}}, j = 1, \dots, d)$ should also be obtained. To reduce the computational cost, a semi-analytical design sensitivity analysis method based on PC was developed. Based on Eqs. (8), (9), and (3), the design sensitivity of the mean (μ_y) and standard deviation (σ_y) of the output response with respect to the design variable μ_{x_j} can be calculated as follows.

$$\frac{d\mu_y}{d\mu_{x_j}} = \frac{db_0}{d\mu_{x_j}} = \frac{d(E[y\Phi_0(\xi)]/E[\Phi_0^2(\xi)])}{d\mu_{x_j}} \quad (10)$$

$$\begin{aligned} \frac{d\sigma_y}{d\mu_{x_j}} &= \frac{d \left(\sum_{i=1}^P b_i^2 E[\Phi_i^2(\xi)] \right)^{1/2}}{d\mu_{x_j}} \\ &= \left(\sum_{i=1}^P b_i^2 E[\Phi_i^2(\xi)] \right)^{-1/2} \left(\sum_{i=1}^P b_i E[\Phi_i^2(\xi)] \frac{db_i}{d\mu_{x_j}} \right) \\ &= \left(\sum_{i=1}^P \frac{E[y\Phi_i(\xi)]^2}{E[\Phi_i^2(\xi)]} \right)^{-1/2} \left(\sum_{i=1}^P \frac{E[y\Phi_i(\xi)]}{E[\Phi_i^2(\xi)]} \frac{dE[y\Phi_i(\xi)]}{d\mu_{x_j}} \right) \end{aligned} \quad (11)$$

Fig. 1 Flowchart of the PC-based robust optimization with gradient-based optimization algorithm



Equations (10) and (11) can be further expressed as the following equations:

$$\frac{d\mu_y}{d\mu_x} = \frac{1}{E[\Phi_0^2(\xi)]} \left(\sum_{i_j=1}^{m_j} \frac{\partial E[y\Phi_0(\xi)]}{\partial x_{j,i_j}} \frac{dx_{j,i_j}}{d\mu_x} + \sum_{i_j=1}^{m_j} \frac{\partial E[y\Phi_0(\xi)]}{\partial \omega_{j,i_j}} \frac{d\omega_{j,i_j}}{d\mu_x} \right) \tag{12}$$

$$\frac{d\sigma_y}{d\mu_x} = \left(\sum_{i=1}^P \frac{E[y\Phi_i(\xi)]^2}{E[\Phi_i^2(\xi)]} \right)^{-1/2} \left(\sum_{i=1}^P \frac{E[y\Phi_i(\xi)]}{E[\Phi_i^2(\xi)]} \left(\sum_{i_j=1}^{m_j} \frac{\partial E[y\Phi_i(\xi)]}{\partial x_{j,i_j}} \frac{dx_{j,i_j}}{d\mu_x} + \sum_{i_j=1}^{m_j} \frac{\partial E[y\Phi_i(\xi)]}{\partial \omega_{j,i_j}} \frac{d\omega_{j,i_j}}{d\mu_x} \right) \right) \tag{13}$$

where x_{j,i_j} and ω_{j,i_j} are the node and weight coefficients in the j th dimension, respectively, which correspond to the random variable x_j .

In Eqs. (12) and (13), given that the value of the weight ω_{j,i_j} is only determined by the number of nodes m_j and distribution of x_j , $\frac{d\omega_{j,i_j}}{d\mu_x} = 0$. Moreover, given that the nodes x_{j,i_j} ($i_j = 1, \dots, m_j$) in each dimension are centered around the mean μ_x with fixed spacing, $\frac{dx_{j,i_j}}{d\mu_x} = 1$. Therefore, the critical task is the calculation of the partial derivative $\frac{\partial E[y\Phi_i(\xi)]}{\partial x_{j,i_j}}$.

3.1 Design sensitivity analysis method based on PC-FFNI

When FFNI is used for the PC coefficient calculation based on Eq. (4), the partial derivatives $\frac{\partial E[y\Phi_i(\xi)]}{\partial x_{j,i_j}}$ in Eqs. (12) and (13) can be expressed as follows.

$$\begin{aligned} \frac{\partial E[y\Phi_i(\xi)]}{\partial x_{j,i_j}} &= \sum_{i_1=1}^{m_1} \omega_{1,i_1} \cdots \sum_{i_{j-1}=1}^{m_{j-1}} \omega_{j-1,i_{j-1}} \sum_{i_{j+1}=1}^{m_{j+1}} \omega_{j+1,i_{j+1}} \cdots \\ &\times \cdots \sum_{i_D=1}^{m_D} \omega_{D,i_D} \frac{\partial y(x_{1,i_1}, \dots, x_{D,i_D})}{\partial x_{j,i_j}} \cdot \omega_{j,i_j} \\ &\times \Phi_i(\xi) \end{aligned} \tag{14}$$

The partial derivative with respect to the node in Eq. (14) can be approximately calculated based on the data (nodes) employed for the previous statistical moment estimation using the finite difference strategy, without the requirement of additional function evaluations. Using the two-dimensional node as an example for illustration, it is assumed that the one-dimensional nodes are located from left to right in each dimension by their sequence numbers (i.e., the subscripts). Based on the three-point Lagrange interpolation polynomial, the partial derivative at any location can be calculated using the forward, center, and backward difference methods as follows.

Three adjacent nodes l_1, l_2 , and l_3 with their corresponding function response values g_1, g_2 , and g_3 are selected, based on which two approximate partial derivatives $r_1 = \frac{g_2-g_1}{l_2-l_1}$ and $r_2 = \frac{g_3-g_2}{l_3-l_2}$ that correspond to nodes $l'_1 = \frac{l_1+l_2}{2}$ and $l'_2 = \frac{l_2+l_3}{2}$, respectively, can be obtained. The partial derivatives r_a at any location l_a can then be approximately calculated using the Lagrange linear interpolation polynomial (Abramowitz et al. 1966), as follows:

$$r_a = \frac{l_a-l'_2}{l'_1-l'_2} r_1 + \frac{l_a-l'_1}{l'_2-l'_1} r_2 \tag{15}$$

According to the locations of nodes l_1, l_2 , and l_3 , there are three possible cases with respect to the calculation of Eq. (15).

Case 1: when $i_j = 1$ ($j = 1, \dots, d$), as shown in Fig. 2a, let $l_1 = x_{j,1}, l_2 = x_{j,2}, l_3 = x_{j,3}$, and $l_a = l_1$. The partial derivative can be approximated by the forward difference method based on Eq. (15), as follows.

$$\begin{aligned} \frac{\partial y(x_{1,i_1}, \dots, x_{D,i_D})}{\partial x_{j,i_j}} &= \frac{l_1-l_2+l_3}{l_1+l_2-l_2+l_3} \cdot \frac{g_2-g_1}{l_2-l_1} + \frac{l_1-l_1+l_2}{l_2+l_3-l_1+l_2} \cdot \frac{g_3-g_2}{l_3-l_2} \\ &= \frac{l_1-l_2+l_1-l_3}{l_1-l_3} \cdot \frac{g_2-g_1}{l_2-l_1} + \frac{l_1-l_2}{l_3-l_1} \cdot \frac{g_3-g_2}{l_3-l_2} \\ &= \frac{2h_1+h_2}{h_1+h_2} \cdot \frac{g_2-g_1}{h_1} + \frac{l_1-l_2}{h_2} \cdot \frac{g_3-g_2}{h_2} \\ &= \frac{h_1}{h_1+h_2} \cdot \frac{g_2-g_1}{h_1} + \frac{h_1+h_2}{h_2} \cdot \frac{g_2-g_1}{h_1} \end{aligned} \tag{16}$$

Case 2: when $i_j = m_j$, as shown in Fig. 2b, let $l_1 = x_{j,i_j-2}, l_2 = x_{j,i_j-1}, l_3 = x_{j,i_j}$, and $l_a = l_3$. Similar to the derivation of Eq. (16), the partial derivative can be approximated using the backward difference method as follows.

$$\frac{\partial y(x_{1,i_1}, \dots, x_{D,i_D})}{\partial x_{j,i_j}} = \frac{g_3-g_1}{h_1+h_2} + \frac{g_3-g_2}{h_2} - \frac{g_2-g_1}{h_1} \tag{17}$$

Case 3: when $i_j \neq 1$ and $i_j \neq m_j$, as shown in Fig. 2c, let $l_1 = x_{j,i_j-1}, l_2 = x_{j,i_j}, l_3 = x_{j,i_j+1}$, and $l_a = l_2$. The partial derivative can be approximated using the central difference method as follows.

$$\frac{\partial y(x_{1,i_1}, \dots, x_{D,i_D})}{\partial x_{j,i_j}} = -\frac{g_3-g_1}{h_1+h_2} + \frac{g_3-g_2}{h_2} + \frac{g_2-g_1}{h_1} \tag{18}$$

In Eqs. (16)–(18), $h_1 = l_2 - l_1$, $h_2 = l_3 - l_2$, and

$$\begin{aligned} g_1 &= y(x_{1,i_1}, \dots, x_{j,1}, \dots, x_{D,i_D}) \\ g_2 &= y(x_{1,i_1}, \dots, x_{j,2}, \dots, x_{D,i_D}) \\ g_3 &= y(x_{1,i_1}, \dots, x_{j,3}, \dots, x_{D,i_D}) \end{aligned} \tag{19}$$

For higher-dimensional cases ($D \geq 3$), the partial derivative can be easily derived for each dimension using the abovementioned methods. To apply the proposed design

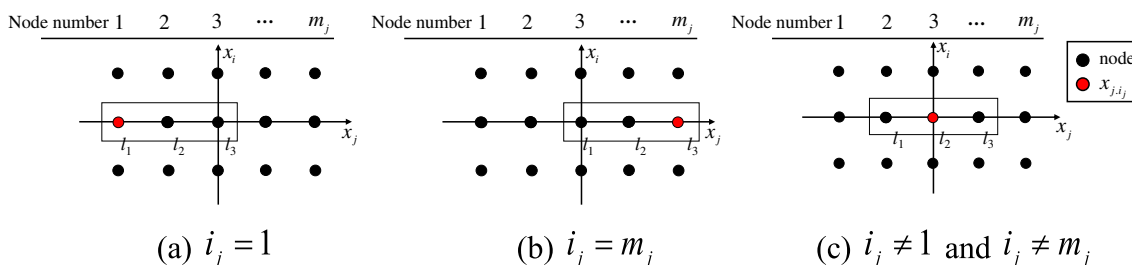


Fig. 2 Calculation of partial derivatives with FFNI nodes

sensitivity analysis method, the number of one-dimensional nodes for each dimension is required to be $m_j \geq 3$. This requirement is satisfied for practical problems using the PC method with FFNI. It should be noted that the design sensitivity obtained by the proposed method is theoretically inaccurate when the performance function $g(x)$ is highly nonlinear. However, the local sensitivity requires calculation within a very small design region; thus, the error induced by the Lagrange linear interpolation is not excessively large, as verified in the comparative studies on robust optimization, which are presented further in this paper.

3.2 Design sensitivity analysis method for PC-SGNI

When SGNI is employed for the PC coefficient calculation, the partial derivatives in Eqs. (12) and (13) can be calculated as follows, based on Eq. (7).

$$\frac{\partial E[y\Phi_0(\xi)]}{\partial x_{j,i_j}} = \sum_{i_1, i_2, \dots, i_D \in \Omega} (-1)^{Q-|i|} \binom{D-1}{Q-|i|} \left(\sum_{i_1=1}^{m_1} \omega_{1,i_1} \dots \sum_{i_D=1}^{m_D} \omega_{D,i_D} \frac{\partial y(x_{1,i_1}, \dots, x_{D,i_D})}{\partial x_{j,i_j}} \cdot \Phi_i(\xi) \right) \quad (20)$$

where Ω represents the collection of all the multi-index combinations $\{i_1, \dots, i_D\}$ that satisfy $K + 1 \leq |i| \leq Q$; and x_{j,i_j} and ω_{j,i_j} are the node and weight coefficients in the j th dimension, respectively, which correspond to one small-size direct full tensor product.

The partial derivative with respect to the node in Eq. (20) can be calculated in the same manner as the FFNI. However, given that SGNI employs the special tensor product, which is a combination of certain small-size direct full tensor products, for the removal of several unimportant integration nodes, the number of one-dimensional nodes for each dimension in the small-size full tensor product may be $m_j = 1, 2, \dots$, which cannot satisfy $m_j \geq 3$. Therefore, minor modifications should be made. Using the two-dimensional node with $K=2$ as an example for illustration, five groups of small-size full tensor products were generated using SGNI (see Fig. 3). The partial derivative with respect to x_j can be calculated with respect to the following two cases.

Case 1: when there are one or two nodes present in the j th dimension x_j for a given group, as illustrated in Fig. 3 (Case 1). The partial derivative cannot be calculated in this case. Several additional nodes based on the original response function are employed for the calculation of the partial derivative, and the induced number of function evaluations is large, especially for higher-dimensional problems. Therefore, the response values of the added nodes are approximately calculated by the PC model, to reduce the computational cost. The partial derivative is then calculated in the same manner as that of FFNI presented above. Given that the PC method is a metamodel in the stochastic domain, such an approximation of the original function response on a few nodes during the design sensitivity analysis calculation would definitely induce some error especially when nonlinear terms with higher polynomial order are involved in the response function. However, such error can be reduced to some extent by increasing the order of PC model. Here, the added nodes are selected as the Gaussian quadrature nodes to ensure a high accuracy.

Case 2: when there is a minimum of three nodes in the j th dimension, as shown in Fig. 3 (Case 2). The calculation procedure of the partial derivative is the same as that of the FFNI.

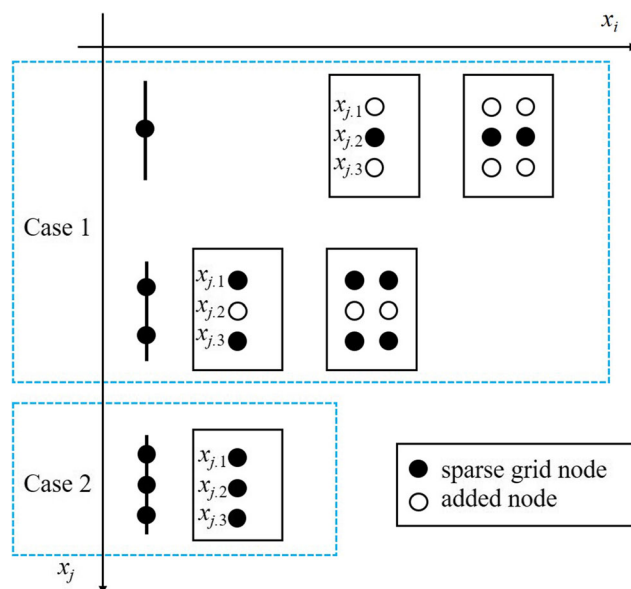


Fig. 3 Calculation of partial derivatives with SGNI nodes

Compared with FFNI, the application of SGNI to design sensitivity analysis by the proposed method would clearly save the number of functional calls and relieve the curse of dimensionality, which however would reduce the accuracy. As fewer sample points are employed, the accuracy of PC model by SGNI is reduced, which would further reduce the accuracy of statistical moment calculation of output response as well as output response prediction, and finally decrease the accuracy of design sensitivity analysis.

For certain larger-scale problems with very high-dimensional random inputs, adaptive methods have been developed by removing several unimportant orthogonal polynomials from the full PC model, or by the adaptive collocation of the integration nodes during the calculation of the PC coefficients for the reduction of the computational cost (Blatman and Sudret 2011; Perkó et al. 2014; Lucor and Karniadakis 2004). In this case, the design sensitivity is calculated using the developed PC model in the same manner without any modifications. In this paper, the Gaussian-type quadrature formula was employed for constructing PC model. However, due to its high flexibility and low computational cost, the least square estimation method is increasingly applied to the PC method for uncertainty propagation and robust optimization. Therefore, it is necessary to implement a design sensitivity analysis for the least square estimation-based PC method for robust optimization in future work.

4 Comparative studies

To demonstrate the effectiveness of the proposed semi-analytical design sensitivity analysis method, the evaluation of the design sensitivity of two mathematical examples is presented in Section 4.1, followed by the robust optimization of three mathematical examples using the proposed design sensitivity method in Section 4.2. The order of the PC models was set as $p = 2$, the number of one-dimensional Gaussian

quadrature nodes for the FFNI was set as $m = 3$, and the accuracy level for the SGNI was set as $K = 2$. Based on the data employed for the statistical moment estimation using the PC method, the design sensitivity was derived for the PC model using the proposed semi-analytical DSA method (denoted by DSA-PC). The traditional finite difference method (FDM) in conjunction with the PC method was also employed for comparison (denoted by FDM-PC), in which the PC method was used for the statistical moment calculation, and the FDM for the sensitivity analysis, by carrying out uncertainty propagation (UP) with the PC method at two points adjacent to the current design point. For the FDM, to ensure the accuracy as much as possible, the central difference method was used to calculate the design sensitivity at x_j in Section 4.1.

$$\frac{\partial y(x)}{\partial x_j} = \frac{y(x_1, \dots, x_j + \Delta, \dots, x_d) - y(x_1, \dots, x_j - \Delta, \dots, x_d)}{2\Delta} \tag{21}$$

The step size clearly has a substantial impact on the calculated partial derivative. Therefore, before calculating design sensitivity, a step convergence test that calculates design sensitivities with different step sizes was conducted, based on which a suitable step size was selected for each example in Section 4.1; while for robust optimization in Sections 4.2 and 4.3, as the optimization toolbox in MATLAB was employed, Δ and the finite difference method (central/back/forward-difference) were automatically assigned by the optimization algorithm in MATLAB. The design sensitivity was simultaneously calculated by the FDM based on the estimated statistical moments with the MCS (denoted by FDM-MCS), for which the MCS was employed for the statistical moment estimation, and the FDM for the sensitivity analysis, by carrying out UP with MCS at two points adjacent to the current design point. The results generated by FDM-MCS were employed as the reference values. The total number of function calls (N) required by the moment estimation and design sensitivity analysis was compared. The sequential quadratic

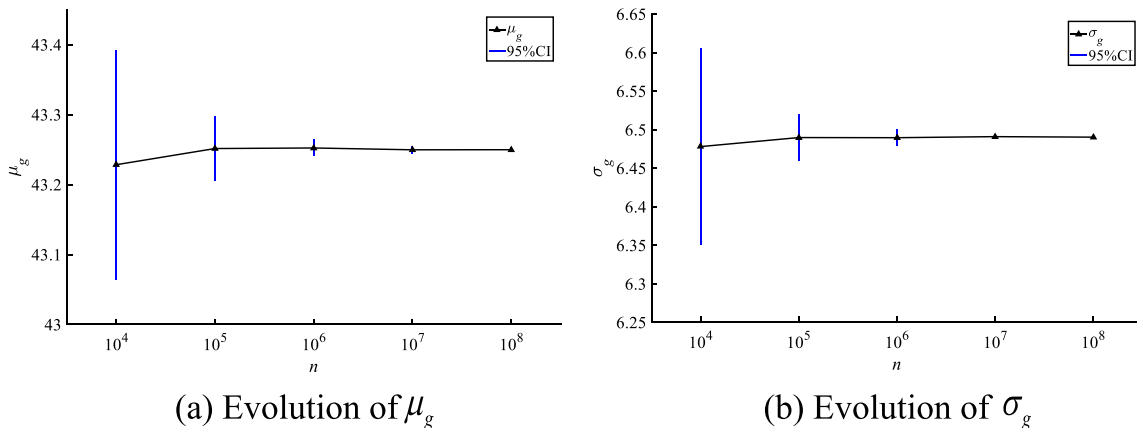


Fig. 4 Convergence test for FDM-MCS with different number of sample points

Table 1 Results of sensitivity analysis for Example 1

	$d\mu_g/d\mu_{x_1}$	$d\mu_g/d\mu_{x_2}$	$d\sigma_g/d\mu_{x_1}$	$d\sigma_g/d\mu_{x_2}$
DSA-PC	2.0000	12.0000	0.6163	0.9244
FDM-PC	2.0000	12.0000	0.6163	0.9244
FDM-MCS	2.0021	12.0056	0.6180	0.9276

programming (SQP) provided in MATLAB was employed as the optimization algorithm considering its high efficiency and good performance. As SQP is a gradient-based optimization algorithm, a local optimal solution may be obtained if a poor initial point is provided. Therefore, as is commonly done in practice, optimization with different starting points is tested to obtain the optimal solutions, among which the best feasible one is selected as the final optimal solution.

4.1 Design sensitivity analysis of mathematical examples

A two-dimensional problem with normal distributed random inputs was first tested:

$$g(x_1, x_2) = (x_1 - 3)^2 + (x_2 + 1)^2 + 5 \tag{22}$$

$x_1 \sim N(4, 1), x_2 \sim N(5, 0.25)$

The FFNI was employed to calculate the PC coefficients. The convergence test for MCS with different number of sample points (n) was firstly done for this example to determine a suitable number of sample points (n), and the 95% confidence interval that was estimated by running 20 times the MCS was calculated. The results of the convergence test were shown in Fig. 4. It was found that when $n = 10^7$, the confidence interval was very small and the mean and standard deviation tended to be constants, indicating that the results are accurate enough and very close to the real values. Therefore, to ensure the effectiveness of comparison, MCS with $n = 10^7$ were

conducted to estimate statistical moments for FDM-MCS. For all the rest mathematical examples (Sections 4.1 and 4.2), the same convergence test was also done for FDM-MCS to determine a suitable n for MCS firstly. For space limit, the results of the convergence test for the rest examples were not shown. The step size was set as $\Delta = 10^{-4}$ considering accuracy after the step convergence test.

The results obtained by the three approaches (DSA-PC, FDM-PC, FDM-MCS) were shown in Table 1. The proposed DSA-PC method was found to generally produce the same results as the FDM-PC, which were similar to those generated by the FDM-MCS. However, the DSA-PC required significantly fewer function calls than FDM-PC (9 vs. 36), as it carried out no additional function calls to calculate the design sensitivity.

A higher-dimensional problem ($d = 10$) with normal distributed random inputs was then tested to verify the effectiveness of the proposed method in a higher-dimensional design sensitivity analysis. The step size was set as $\Delta = 10^{-4}$.

$$f(x_1, \dots, x_{10}) = \frac{(x_1 + x_2 + x_3)^2 + 3x_4^3 + 4(x_5 \times x_6)}{5(x_7 + x_8)(x_9 + x_{10})}$$

$x_1 \sim N(3, 4), x_2 \sim N(4, 1), x_3 \sim N(5, 4), x_4 \sim N(4, 4), x_5 \sim N(3, 1)$
 $x_6 \sim N(6, 1), x_7 \sim N(8, 4), x_8 \sim N(6, 4), x_9 \sim N(9, 1), x_{10} \sim N(10, 4)$

(23)

For the DSA-PC and FDM-PC, the PC coefficients were calculated by SGNI, given that the dimension was relatively high. Similarly, for FDM-MCS, the number of sample points

Table 2 Results of sensitivity analysis for Example 2

Sensitivity	DSA-PC	FDM-PC	FDM-MCS	Sensitivity	DSA-PC	FDM-PC	FDM-MCS
$d\mu_g/d\mu_{x_1}$	0.0192	0.0192	0.0192	$d\sigma_g/d\mu_{x_1}$	0.0023	0.0024	0.0022
$d\mu_g/d\mu_{x_2}$	0.0192	0.0192	0.0192	$d\sigma_g/d\mu_{x_2}$	0.0023	0.0024	0.0022
$d\mu_g/d\mu_{x_3}$	0.0192	0.0192	0.0192	$d\sigma_g/d\mu_{x_3}$	0.0023	0.0024	0.0022
$d\mu_g/d\mu_{x_4}$	0.1305	0.1437	0.1457	$d\sigma_g/d\mu_{x_4}$	0.1060	0.1264	0.1278
$d\mu_g/d\mu_{x_5}$	0.0192	0.0192	0.0192	$d\sigma_g/d\mu_{x_5}$	0.0016	0.0017	0.0017
$d\mu_g/d\mu_{x_6}$	0.0096	0.0096	0.0096	$d\sigma_g/d\mu_{x_6}$	0.0011	0.0011	0.0012
$d\mu_g/d\mu_{x_7}$	-0.0343	-0.0353	-0.0358	$d\sigma_g/d\mu_{x_7}$	-0.0301	-0.0325	-0.0366
$d\mu_g/d\mu_{x_8}$	-0.0343	-0.0353	-0.0358	$d\sigma_g/d\mu_{x_8}$	-0.0301	-0.0325	-0.0366
$d\mu_g/d\mu_{x_9}$	-0.0241	-0.0243	-0.0243	$d\sigma_g/d\mu_{x_9}$	-0.0190	-0.0206	-0.0207
$d\mu_g/d\mu_{x_{10}}$	-0.0241	-0.0243	-0.0243	$d\sigma_g/d\mu_{x_{10}}$	-0.0190	-0.0206	-0.0207

Table 3 Optimization results of Example 1

	DSA-PC	FDM-PC	FDM-MCS
$\mu_{x_1}^*$	3.4447	3.4044	3.3582
$\mu_{x_2}^*$	4.9995	5.0004	5.0037
F^c	0.07774	0.07617	0.07559
G^c	0.2968	0.2580	0.2149
F	0.07772	0.07618	/
G	0.2971	0.2577	/
$N(a/b)$	117/117	162/162	6.8e8/6.8e8

a/b: *a* represents the number of calls to *f*, and *b* represents the number of calls to *g*

of MCS was set as $n = 10^7$ by the convergence test. The results obtained by DSA-PC, FDM-PC, and FDM-MCS were shown in Table 2. The proposed DSA-PC method was found to basically produce very similar results to the FDM-PC, and the results of both methods were similar to those of the FDM-MCS. However, the FDM-PC generally exhibited a slightly higher accuracy than the DSA-PC. This can be attributed to the use of SGNI for the calculation of the PC coefficients for the DSA-PC, in which several response values were approximated by the PC method instead of referring to the original response function, to reduce the computational cost. Thus, a degree of error was presented into the design sensitivity analysis, which only had a slight influence, given that the approximation by the PC was relatively accurate. However, the design sensitivity with respect to x_4 produced by the DSA-PC was clearly less accurate than that of the FDM-MCS; the difference of DSA-PC relative to FDM-PC is up to 15.43%. This is because the response function was more nonlinear with respect to x_4 , and the PC model with order $p = 2$ would produce relatively large error on response prediction, which would induce large error on sensitivity calculation for x_4 when SGNI is employed. The FDM-PC exhibited a higher accuracy than the DSA-PC, but the DSA-PC required a significant lower number of function calls (231 vs. 4620) due to the employment of the proposed design sensitivity method. To reduce such error, the order of PC model is increased from $p = 2$ to $p = 3$ and the design sensitivities were calculated again. As the errors of sensitivities with respect to x_4 were relatively large with $p = 2$, here only results with respect to x_4 are shown: $d\mu_g/d\mu_{x_4} = 0.1419$, $d\sigma_g/d\mu_{x_4} = 0.1197$. Clearly, with the

increase of PC order, the accuracy of design sensitivity by the proposed DSA-PC method is improved. The results from the two examples demonstrate the accuracy and effectiveness of the proposed semi-analytical design sensitivity method.

4.2 Robust optimization of mathematical examples

4.2.1 Example 1

$$\text{Min}_{\mu_{x_1}, \mu_{x_2}} F = \frac{\sigma_f}{15} \tag{24}$$

$$\text{s.t. } G = \mu_g - 3\sigma_g \geq 0$$

$$1 \leq \mu_{x_1} \leq 10, 1 \leq \mu_{x_2} \leq 10$$

$$\text{where } f(x_1, x_2) = (x_1 - 4)^3 + (x_1 - 3)^4 + (x_2 - 5)^2 + 10$$

$$g = x_1 + x_2 - 6.45$$

The two random design variables x_1 and x_2 were assumed to be normally distributed with the standard deviation as $\sigma = 0.4$. The robust optimization was carried out using the DSA-PC, FDM-PC, and FDM-MCS (with $n = 10^7$ for MCS); and FFNI was employed to calculate the PC coefficients. In optimization, the starting point was set as $x_0 = [3.0, 5.0]$. Once optimization is completed, the confirmed results (F^c and G^c) were obtained by substituting the obtained optimal design variables into the original functions (f and g), respectively, using the MCS method ($n = 10^7$) with the consideration of the same uncertainties. The optimal design variables, the confirmed results, and the total number of function calls of the three methods were presented in Table 3.

Table 4 Distribution parameters for input random variable and parameters

Random var. and para.	Distribution	Mean	Standard deviation
x_1	Normal	μ_{x_1}	$0.02 \times \mu_{x_1}$
x_2	Normal	μ_{x_2}	$0.02 \times \mu_{x_2}$
ρ	Normal	1	0.2
Q	Normal	8	0.2
S	Normal	10.5	0.25

Table 5 Optimization results of Example 2

	DSA-PC	FDM-PC	FDM-MCS
$\mu_{x_1}^*$	0.2000	0.2000	0.2000
$\mu_{x_2}^*$	0.1000	0.1000	0.1005
F^c	0.02015	0.02015	0.02015
G_1^c	0.01803	0.01803	0.01813
G_2^c	0.01825	0.01825	0.01834
F	0.02015	0.02015	/
G_1	0.01803	0.01803	/
G_2	0.01825	0.01825	/
$N(a/b)$	297/891	486/1458	5.5e8/5.5e8

a/b : a represents the number of f calls, and b represents the number of g_1 or g_2 calls

As can be seen from Table 3, the three methods obtained feasible optimal solutions with the satisfied constraint G . Moreover, the optimal design variables ($\mu_{x_1}^*, \mu_{x_2}^*$), in addition to the confirmed objectives of the FDM-PC and DSA-PC, were slightly different. However, they were very similar to those generated by the FDM-MCS. As the value of constraint $G = \mu_g - 3\sigma_g \geq 0$ depends on μ_g , and the magnitude order of the constraint function G is small (10^{-1}), a small difference in $\mu_{x_1}^*$ or $\mu_{x_2}^*$ would induce relatively large difference in G^c (FDM-MCS: 0.2149; DSA-PC: 0.2968; FDM-PC = 0.2580). Therefore, the error of G^c seems substantially high. To reduce such error, a higher PC order for constraint function g can be employed. However, for practical optimization problems, it is required to find an effective design solution that is feasible (i.e. satisfies the constraints), and it is the optimal design solution that is cared about by users, not the exact values of constraints (only need to be feasible). Moreover, the total number of function calls (N) during optimization was reduced by 27.8% using the proposed DSA-PC method when compared with the FDM-PC.

Table 6 Distribution parameters of input random variable

Random var. and para.	Range of $\mu_{x_1}, \dots, 8$	Distribution	Mean/A	Std/B
x_1	[1, 10]	Normal	μ_{x_1}	0.016
x_2	[2, 10]	Normal	μ_{x_2}	$0.1 \times \mu_{x_2}$
x_3	[1, 10]	Uniform	$\mu_{x_3} + 0.1$	$\mu_{x_3} - 0.1$
x_4	[1, 8]	Normal	μ_{x_4}	0.15
x_5	[1, 6]	Uniform	$\mu_{x_5} + 0.15$	$\mu_{x_5} - 0.15$
x_6	[1, 7]	Uniform	$\mu_{x_6} + 0.2$	$\mu_{x_6} - 0.2$
x_7	[5, 15]	Normal	μ_{x_7}	0.04
x_8	[6, 14]	Uniform	$\mu_{x_8} + 0.05$	$\mu_{x_8} - 0.05$

4.2.2 Example 2

$$\text{Min}_{\mu_{x_1}, \mu_{x_2}} F = 0.5 \frac{\mu_f}{10} + 0.5 \frac{\sigma_f}{2} \tag{25}$$

$$\text{s.t. } G_1 = \mu_{g_1} - 3\sigma_{g_1} \geq 0$$

$$G_2 = \mu_{g_2} - 3\sigma_{g_2} \geq 0$$

$$0.2 \leq \mu_{x_1} \leq 20, 0.1 \leq \mu_{x_2} \leq 1.6$$

where $f(x_1, x_2) = \rho x_1 \sqrt{1 + x_2^2}$

$$g_1 = x_1 x_2 - \frac{0.06 Q x_1 x_2}{\sqrt{65} S} \sqrt{1 + x_2^2} \times (8x_1 + 1) \geq 0$$

$$g_2 = x_1 x_2 - \frac{0.06 Q x_1 x_2}{\sqrt{65} S} \sqrt{1 + x_2^2} \times (8x_1 - 1) \geq 0$$

The distribution parameters of the random variables and random parameters are shown in Table 4. The robust design optimization was carried out using the DSA-PC, FDM-PC, and FDM-MCS ($n = 10^7$ for MCS). The FFNI was employed to calculate the PC coefficients for the moment estimation. In optimization, the starting point was set as $x_0 = [3.0, 0.1]$.

Table 5 presents the optimization results, in which F^c , G_1^c , and G_2^c represent the results confirmed using the MCS method ($n = 10^7$). As can be seen from Table 5, the results obtained by the FDM-PC and DSA-PC were approximately equal, which were in good agreement with those obtained by the FDM-MCS. Moreover, when compared with the FDM-PC, the function evaluations of the original objective f and constraints g for the DSA-PC were both reduced by 38.9%. The proposed DSA-PC method can significantly reduce the computational cost while maintaining an adequate accuracy of design sensitivity.

4.2.3 Example 3

$$\text{Min}_{\mu_{x_1}, \dots, \mu_{x_8}} F = 1.5\mu_f + 0.4\sigma_f \tag{26}$$

$$\text{s.t. } G_1 = \mu_{g_1} - 3\sigma_{g_1} \geq 0$$

$$\text{where } f(x_1, \dots, x_8) = \left(\frac{(x_1-5)}{2} + \frac{(x_2-4)}{3} - 3 \right)^2 + (x_3-3)^4 + (x_3 + x_4 - 4)^2 + (x_5 - 4)^2 + (4x_6 - 6)^4 + (x_7 - 7)^2 + (x_8 - 8)^2$$

$$g = 2 + (x_3 + x_4 - 4)(x_1 + x_2 - 14)$$

Table 7 Optimization results of Example 3

	DSA-PC	FDM-PC	FDM-MCS
$\mu_{x_1}, \mu_{x_2}, \mu_{x_3}, \mu_{x_4}, \mu_{x_5}, \mu_{x_6}, \mu_{x_7}, \mu_{x_8}$	10.0000, 5.4793, 3.0101, 1.0000, 4.0007, 1.5009, 7.0005, 8.0019	10.0000, 5.4631, 2.9998, 1.0002, 4.0000, 1.5000, 7.0000, 8.0000	10.0000, 5.4567, 3.0000, 1.0000, 4.0000, 1.5000, 7.0000, 8.0000
F^C	0.2769	0.2764	0.2764
G^C	1.2538	1.2470	1.2498
F	0.2703	0.2696	/
G	1.2449	1.2469	/
$N(a/b)$	7497/2205	13,158/3870	1.76e9/1.76e9

a/b: *a* represents the number of *f* calls, and *b* represents the number of *g* calls

The distribution parameters of the random variables are shown in Table 6, in which “Range” represents the design region of the design variable during optimization; “Std” denotes the standard deviation of the distribution; and *A* and *B* represent the upper and lower limits of the uniform distributed random design variable, respectively. The robust design optimization was carried out using the DSA-PC, FDM-PC, and FDM-MCS ($n = 10^7$ for MCS). Considering that it was a relatively high-dimensional problem, SGNI was employed to calculate the PC coefficients for the moment estimation. In optimization, the starting point was set as $x_0 = [8.0, 4.0, 2.0, 1.0, 3.0, 1.0, 5.0, 6.0]$.

As can be seen from the optimization results shown in Table 7, the optimal solutions obtained by the FDM-PC and DSA-PC were very similar, but the FDM-PC exhibited a slightly higher accuracy than the DSA-PC. This can be attributed to the use of additional nodes based on the approximation of PC model, for the calculation of a sensitivity term, which introduces a degree of error into the final design sensitivity. Such error does not have large impact on the optimal solution of this example when $p = 2$, even if the highest order of the response functions is 4, demonstrating the effectiveness of approximation with PC for high-dimensional problems in the DSA-PC. However, when compared with the FDM-PC, the DSA-PC reduced the required number of function evaluations of the original objective *f* and constraint *g* by 43.0%.

The proposed DSA-PC method can significantly reduce the computational cost while maintaining an adequate accuracy of design sensitivity.

Further, to investigate the impact of sensitivity estimation error induced by Lagrange linear interpolation on the optimization convergence speed, the evolutions of the objective function (*F*) for the proposed DSA-PC and the FDM-MCS during the optimization process for the three examples above were shown in Figs. 5, 6, and 7, respectively. Three noteworthy observations were made.

Firstly, it was found that with the increase of iteration number, the values of *F* for both methods (DSA-PC and FDM-MCS) for the three examples were generally decreased rapidly, and remained almost unchanged at the end of optimization, indicating that the optimal solutions were convergent. Meanwhile, the values of *F* produced by both approaches were close to each other, especially for Example 2. Generally, only slightly more (1 or 3) iteration numbers were required by DSA-PC.

Secondly, compared with Example 1 and Example 2, more iterations were required for Example 3. The interpretation is as follows. Example 3 was relatively a high-dimensional ($d = 8$) problem; and meanwhile, the SGNI technique was employed for calculating the PC coefficients, and the response values of some added nodes were approximately calculated by the PC model, inducing another source of error in sensitivity

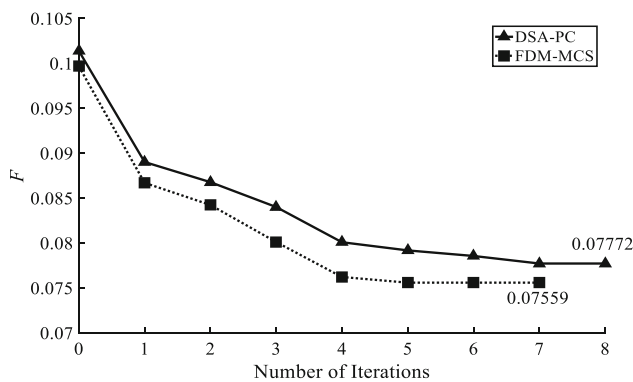


Fig. 5 Evolution of *F* during optimization (Example 1)

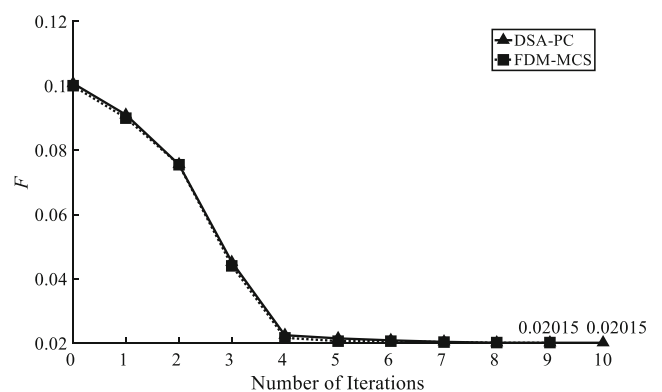


Fig. 6 Evolution of *F* during optimization (Example 2)

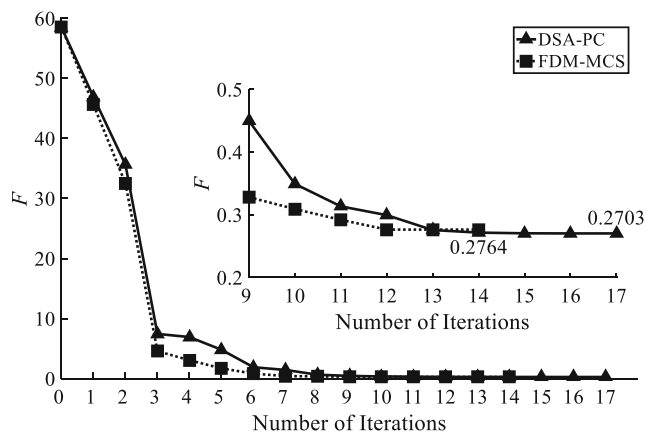


Fig. 7 Evolution of F during optimization (Example 3)

calculation. However, from Fig. 7, it was noticed that F of the proposed DSA-PC method rapidly approached to the optimal solution.

Thirdly, it was also noticed that the difference in F during the whole iteration process produced by both methods for Example 1 was the largest, which was relatively large even at the initial starting design point. The interpretation is that for Example 1, the response function $f(x_1, x_2)$ is relatively more nonlinear and the standard deviation of design variable is relatively large ($\sigma=0.4$, 8% ~ 12% of the mean value), and thus relatively large errors would be induced in the statistical moment calculation using 2-order PC and further in sensitivity analysis. However, from the optimal results shown in Table 3, it was found that the results produced by DSA-PC were close to that of FDM-MCS, which further demonstrated the high accuracy of sensitivity estimation.

All these observations indicated that the sensitivity estimation error induced by Lagrange linear interpolation of the proposed DSA-PC method had a very small impact on the optimization convergence, which was attributed to the high accuracy of sensitivity estimation.

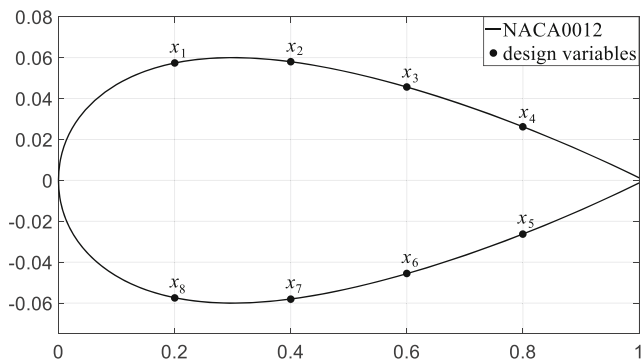


Fig. 8 Airfoil parametric modeling

4.3 Airfoil robust optimization design

The proposed DSA-PC method was applied to an airfoil robust optimization problem, and the NACA0012 airfoil was considered the baseline (Wang et al. 2019b). The objective was the determination of the optimal geometric parameters of the airfoil, to maximize the lift-drag ratio of the airfoil subject to the thickness constraint of the airfoil. The nominal flow condition considered corresponds to the angle of attack $\alpha = 1.55^\circ$, Mach number $M_a = 0.70$ and Reynolds number $Re = 9e6$. It is a typical flight Reynolds number of a commercial aircraft. The B-spline curve (Wang and Liu 2016) with eight control points was employed for the shape parameterization of the airfoil, where the horizontal locations of the eight control points were fixed as $l = [0.2, 0.4, 0.6, 0.8, 0.8, 0.6, 0.4, \text{ and } 0.2]$ and the vertical locations were free to move, as shown in Fig. 8. The original airfoil was considered the baseline with the vertical locations of the eight control points as $z_{\text{baseline}} = [0.0574, 0.0580, 0.0456, 0.0262, -0.0262, -0.0456, -0.0580, \text{ and } -0.0574]$. The vertical locations of the eight control points were considered design variables $x = [x_1, \dots, x_8]$.

Clearly, from the physical mechanism, Re and α have large impacts on the result of CFD simulation. Generally, only one category of fluid viscosity is investigated in a study, and it is impossible to study the impacts of all the different kinematic viscosities. Therefore, as for the impact of Re , we mainly considered the influence of Ma . For robust optimization, all the design variables $x_i (i = 1, \dots, 8)$ were assumed to be uncertain considering the manufacturing error and in accordance with the normal distribution, with the standard deviation as $\sigma_{x_i} = 0.0015$. Meanwhile, M_a and α were assumed to follow uniform distribution with variations ± 0.1 and $\pm 0.5^\circ$ around their nominal values ($\alpha = 1.55$, $M_a = 0.70$), respectively. The airfoil robust optimization was formulated as follows.

Table 8 Grid convergence test on the baseline airfoil

Grid density		Lift-to-drag ratio
Far field	Airfoil boundary	
80	160	21.6941
100	200	24.7979
120	240	26.5889
150	300	28.6412
180	360	28.8166

$$\text{Min}_{\mu_{x_1}, \dots, \mu_{x_8}} F = -\mu_f + 3\sigma_f \tag{27}$$

$$\text{s.t. } G = \mu_{T_w} \geq 0.1$$

$$\mu_{x_i} + 3\sigma_{x_i} \leq 0.005$$

$$\mu_{x_i} - 3\sigma_{x_i} \geq -0.005$$

$$-0.005 \leq \mu_{x_i} \leq 0.005$$

where μ_{T_w} is the mean value of the maximum relative thickness T_w .

Computational fluid dynamics (CFD) was employed for the aerodynamic analysis, to obtain the lift and drag. In addition, the mesh was generated using the CFD pre-processor Gambit software. During the aerodynamic analysis, the k-omega two-equation turbulent model was employed, and Fluent 17.0 was used as the CFD solver. In order to verify the accuracy of the result of CFD simulation, the grid convergence test was firstly performed on the baseline airfoil at the nominal flow condition ($\alpha = 1.55^\circ$, $M = 0.70$, and $Re = 9e6$) and the results were shown in Table 8. From Table 8, it was observed that when the grid density was increased to some value (far field: 150; airfoil boundary: 300), the lift-drag ratio remained almost unchanged. Thus, the number of nodes in the grid far field was set as 150, whereas the number of airfoil boundary nodes was set as 300 in CFD simulation of this study.

Then, the lift (C_l) and drag (C_d) coefficients of the baseline NACA 0012 for the nominal flow condition ($\alpha = 1.55^\circ$, $M = 0.70$ and $Re = 9e6$) calculated by CFD were compared with the experimental data from Harris’s test in the Langley 8-Foot Transonic Pressure Tunnel (Harris 1981), which were shown in Table 9. It was noticed that the CFD results were very close to the experimental data, validating the effectiveness of CFD analysis in this work.

Robust optimization was carried out using the DSA-PC and FDM-PC, and the deterministic optimization (DO) was carried out without considering any uncertainties for comparison. Moreover, SGNI with an accuracy level of $K = 2$ was employed for the PC coefficient calculation. For μ_{T_w} , given that it did not require a computationally complex CFD analysis, it was directly obtained using the MCS method.

The optimal airfoil geometries generated using the DSA-PC, FDM-PC, and DO were shown in Fig. 9. It was observed

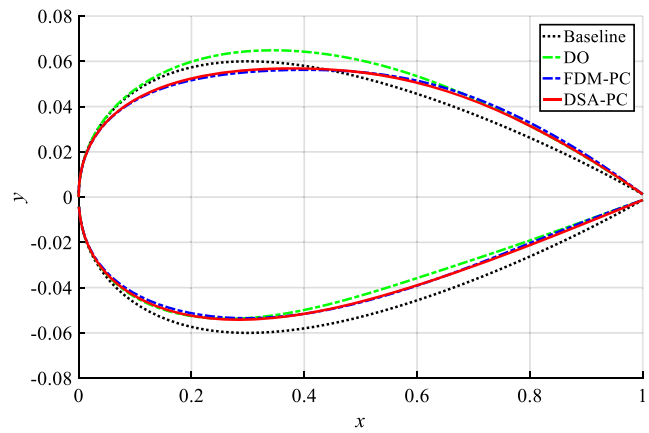


Fig. 9 Optimized and baseline airfoils

that when compared with the original airfoil, the airfoils optimized using the deterministic optimization (DO) and robust optimization (RO) methods were both more bent, and the curvatures of the upper wings were increased, especially for the one produced by DO, which increased the airflow speed. Therefore, the air pressure of the upper wing decreased after optimization, resulting in an increase in the lift coefficient. In addition, the thickness of the leading edge of the optimized airfoils was obviously reduced compared with the original baseline one, which would weaken the shock wave area, and thus reduce the drag coefficient. Finally, an increase in lift-drag ratio was obtained through optimization. Furthermore, the airfoil generated from the robust optimization using the DSA-PC was almost overlapped to that of the FDM-PC. These results demonstrate the effectiveness of the proposed DSA-PC method.

The static pressure contours were shown in Fig. 10. As can be seen from the figure, when compared with the baseline airfoil, the static pressures of the upper surfaces of the airfoils generated by the RO and DO methods were smaller (the area in blue increased after optimization); whereas the static pressures of the lower surface were larger (the area in green increased after optimization). This indicates an increase in the differential pressure between the upper and lower surfaces; thus, there would be an increase in the lift of the airfoil after optimization. Moreover, the static pressure cloud produced by the DSA-PC was very similar to that of the FDM-PC. These results further demonstrate the effectiveness of the proposed DSA-PC method.

The pressure coefficients of the upper and lower surfaces of the airfoil were illustrated in Fig. 11. It could be noticed that compared with the baseline airfoil, the upper surface pressure coefficients of all the optimized airfoils were decreased, while the lower surface pressure coefficients were increased. Meanwhile, the gap between the lower and upper surfaces was increased after optimization, which was the largest for DO. These results indicate that the pressure difference

Table 9 Comparison of CFD results and experimental data

	Experiment	Simulation	Relative error
C_l	0.241	0.235	2.49%
C_d	0.0079	0.0082	3.78%

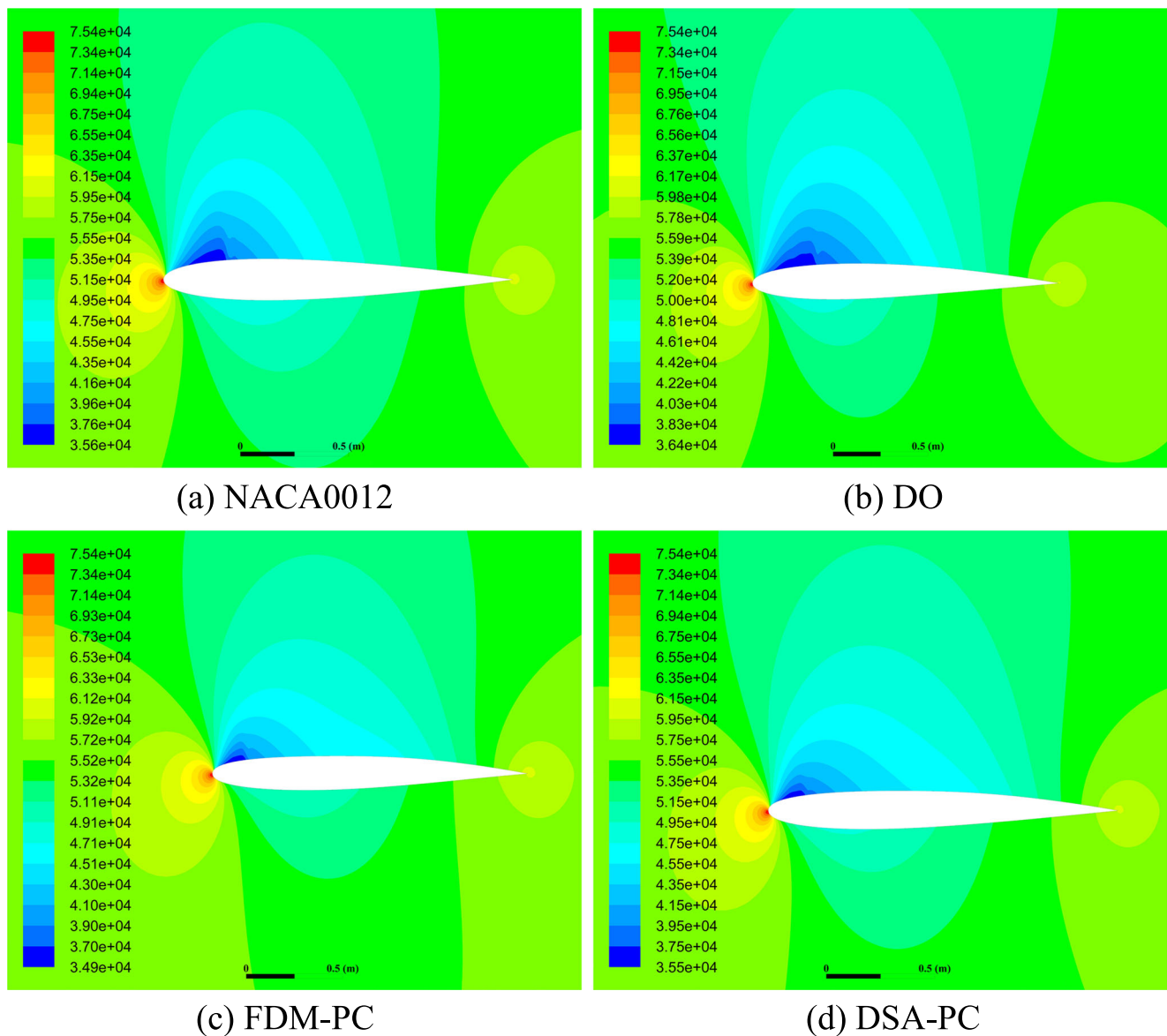


Fig. 10 Static pressure clouds for: (a) baseline of NACA0012, (b) optimized result of deterministic optimization, (c) optimized result of robust optimization with FDM-PC, (d) optimal result of robust optimization with DSA-PC

between the upper and lower surfaces is increased, resulting in an increase in lift after optimization.

The evolution of the objective function (F) during the optimization process was shown in Fig. 12. As the computational cost using MCS-PC (MCS was directly employed for statistical moment calculation) was relatively large, it was not tested for this example and only FDM-PC was compared. It was found that with the increase of iteration number, the values of objective function F of both methods (DSA-PC and FDM-PC) were generally decreased, and remained almost unchanged at the end of the optimization, indicating that the optimal results were convergent. Meanwhile, the values of F produced by DSA-PC and FDM-PC were similar to each other, and slightly more iterations were required by DSA-PC.

These observations showed great agreements with those of the mathematical examples in Section 4.2. For DSA-PC, SGNI was used as it was relatively a high-dimensional problem, in which PC was employed to approximate the response values of some added nodes, and thus more or less errors would be induced. Therefore, a larger iteration number of DSA-PC was required. However, such approximation basically did not have any impact on the obtained performance of the airfoil.

The optimal results of the different methods are shown in Table 10, in which N represents the function evaluations of the CFD for the calculation of the lift-drag ratio during optimization. From Table 10, several noteworthy observations can be made. First, both RO and DO increased the lift-drag ratio of

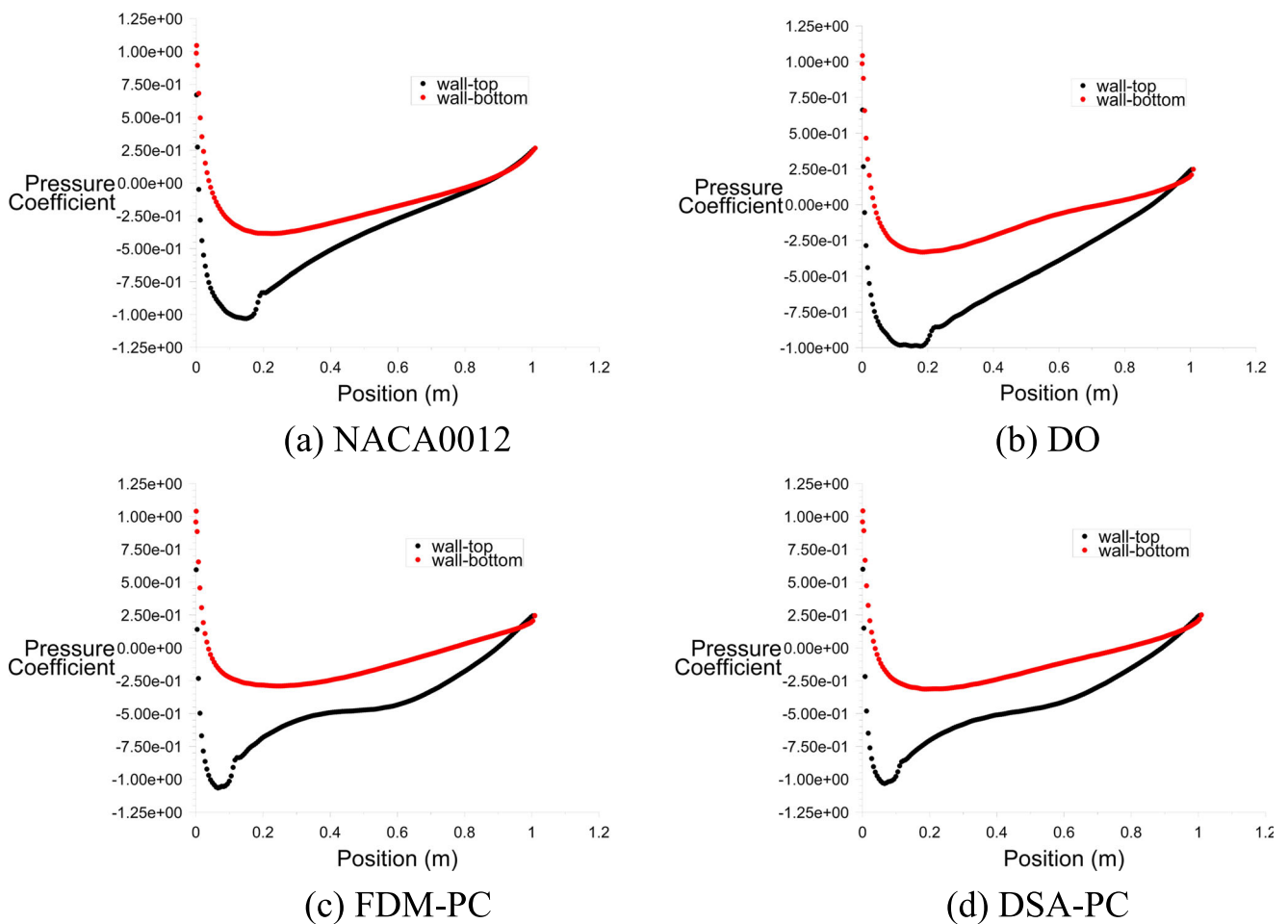


Fig. 11 Pressure coefficient along the upper and lower airfoil surfaces for: (a) baseline of NACA0012, (b) optimal result of deterministic optimization, (c) optimized result of robust optimization with FDM-PC, (d) optimized result of robust optimization with DSA-PC

the airfoil (μ_f) by optimization and satisfied the thickness constraint. This was consistent to the results shown in Figs. 6, 7, and 8, that the lift was increased and the drag was decreased after optimization. Second, the obtained lift-drag ratio (μ_f) of the DO was larger than those of the two ROs; whereas the

standard deviation (σ_f) of the two ROs were significantly smaller, thus exhibiting better robustness with respect to uncertainties. This was consistent with the scenario of robust optimization. Third, the airfoil thickness of the leading edge of the two ROs and DO was obviously reduced compared with that of the original baseline airfoil, and it was decreased more for ROs, which could result in a decrease in the drag coefficient. This perfectly corresponded to the drag

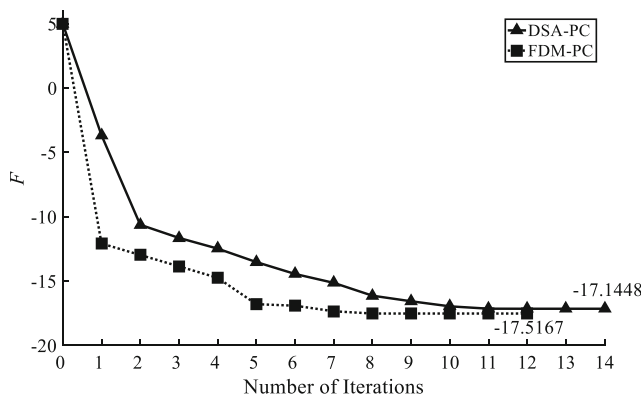


Fig. 12 Evolution of objective function during optimization

Table 10 Optimal results of the airfoil

	Baseline	DSA-PC	FDM-PC	DO
μ_f	22.6946	30.0980	29.6915	34.7129
σ_f	9.3782	4.3419	4.0594	7.6719
μ_{Cl}	2.3977e-1	3.0683e-1	3.0349e-1	3.8403e-1
μ_{Cd}	1.1598e-2	1.0537e-2	1.0507e-2	1.1367e-2
μ_{T_w}	0.1045	0.1020	0.1020	0.1027
N	/	15,246	26,103	315

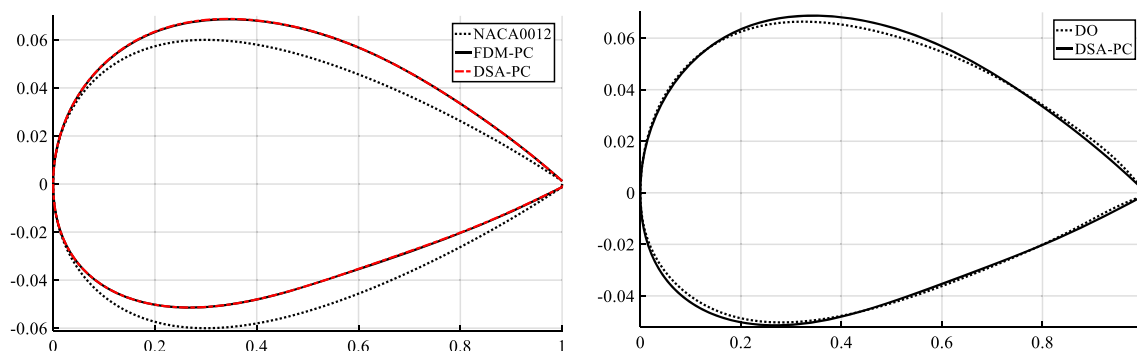


Fig. 13 Optimal airfoils considering manufacturing uncertainty

value μ_{Cd} that it was decreased after optimization. In conclusion, the results provided in Figs. 9, 10, 11, and 12 and Table 10 suggested that the increase in the lift-drag ratio for DO was mainly ascribed to the increase in lift, while it was mainly ascribed to the decrease in drag for ROs. Fourth, the results of the proposed DSA-PC were similar to those of the FDM-PC, whereas the DSA-PC significantly reduced the computational cost (N) by approximately 41.6% due to the employment of the semi-analytical design sensitivity analysis method. All these results presented above demonstrate the effectiveness of the proposed DSA-PC method.

In addition, optimization was also done considering only uncertainties from design variables, of which the optimized airfoils were shown in Fig. 13. It was found that the airfoils generated by DO and ROs did not exhibit clear difference, indicating that the uncertainty in airfoil geometry has small impact on the lift-to-drag ratio. According to all the optimal results provided, we can conclude that Ma and α have large impact on the airfoil performance, which should be taken into account during the airfoil design.

5 Conclusion

In this paper, to reduce the computational cost of the calculation of the local design sensitivity during robust design optimization with computationally expensive response functions and difficulty in applying the adjoint method, a semi-analytical sensitivity method is proposed based on the PC method in conjunction with the Gaussian quadrature formulae. With the proposed method, the local design sensitivity during the optimization iteration is directly derived in the PC model using the nodes for the statistical moment calculation, instead of employing many additional function evaluations. Several mathematical examples and an aerodynamic robust optimization problem were used to demonstrate the effectiveness of the proposed design sensitivity analysis method. The results proved that the proposed method can lead to satisfactory design sensitivities and optimal solutions that are close to

those produced by the traditional finite difference method, while clearly reduce the computational cost of robust optimization. For input random dimension within 10, the proposed method can save up to 43.0% functional calls. The higher of the random input dimension, the more obvious of the computational saving. For problems with highly nonlinear response functions, the proposed method with a low PC order may yield evident errors, which however can be reduced by increasing the PC order. Future work will be focused on developing a design sensitivity analysis approach for the least square estimation-based PC method for robust optimization.

Funding The work was supported by the National Numerical Wind Tunnel Project (grant number NNW2020ZT7-B31) and Science Challenge Project (grant number TZ2018001).

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

Replication of results The results shown in the manuscript can be reproduced. Considering the size limit of the uploaded supplementary material, the codes for two of the mathematical examples for UP (Example 1 in Section 4.1 and Example 2 in Section 4.2) were uploaded as [supplementary material](#). The remaining examples are very easy to implement by changing the response functions and sample points based on the codes provided to obtain the results shown in the manuscript.

References

- Abramowitz M, Stegun IA, Romer RH (1966) Handbook of mathematical functions with formulas, graphs, and mathematical tables. *Phys Today* 19(1):120–121. <https://doi.org/10.1115/1.3625776>
- Beyer HG, Sendhoff B (2007) Robust optimization—a comprehensive survey. *Comput Methods Appl Mech Eng* 196:3190–3218. <https://doi.org/10.1016/j.cma.2007.03.003>
- Blatman G, Sudret B (2011) Adaptive sparse polynomial chaos expansion based on least angle regression. *J Comput Phys* 230(6):2345–2367. <https://doi.org/10.1016/j.jcp.2010.12.021>
- Cheng Q, Wang SW, Yan CC (2016) Robust optimal design of chilled water systems in buildings with quantified uncertainty and reliability

- for minimized life-cycle cost. *Energy Buildings* 126:159–169. <https://doi.org/10.1016/j.enbuild.2016.05.032>
- Cho H, Choi KK, Lee I et al (2016) Design sensitivity method for sampling-based RBDO with varying standard deviation. *ASME J Mech, Des* 138(1):011405. <https://doi.org/10.1115/1.4031829>
- Dodson M, Parks GT (2015) Robust aerodynamic design optimization using polynomial chaos. *J Aircraft* 46(2):635–646. <https://doi.org/10.2514/1.39419>
- Du X, Leifsson L (2019) Optimum aerodynamic shape design under uncertainty by utility theory and metamodeling. *Aerosp Sci Technol* 105464. <https://doi.org/10.1016/j.ast.2019.105464>
- Fisher J, Bhattacharya R (2011) Optimal trajectory generation with probabilistic system uncertainty using polynomial chaos. *J Dyn Sys, Meas, Control* 133(1):014501. <https://doi.org/10.1115/1.4002705>
- Ghisu T, Shahpar S (2017) “Toward affordable uncertainty quantification for industrial problems-part i: theory and validation.” ASME Turbo Expo 2017: Turbomachinery Technical Conference and Exposition, 26–30 June, North Carolina, USA
- Ghisu T, Shahpar S (2018) Affordable uncertainty quantification for industrial problems: application to aero-engine fans. *J Turbomach* 140(6):061005–061005-12. <https://doi.org/10.1115/1.4038982>
- Harris CD (1981) Two-dimensional aerodynamic characteristics of the NACA 0012 airfoil in the Langley 8-Foot Transonic Pressure Tunnel. NASA Ames Research Center, NASA TM 81927
- Janusevskis J, Rodolphe LR (2013) Simultaneous kriging-based estimation and optimization of mean response. *J Global Optim* 55(2):313–336. <https://doi.org/10.1007/s10898-011-9836-5>
- Keshavarzzadeh V, Meidani H, Tortorelli DA (2016) Gradient based design optimization under uncertainty via stochastic expansion methods. *Comput Method Appl M* 306:47–76. <https://doi.org/10.1016/j.cma.2016.03.046>
- Keshavarzzadeh V, Fernandez F, Tortorelli DA (2017) Topology optimization under uncertainty via non-intrusive polynomial chaos expansion. *Comput Method Appl M* 318:120–147. <https://doi.org/10.1016/j.cma.2017.01.019>
- Kumar D, Raisee M, Lacor C (2018) Combination of polynomial chaos with adjoint formulations for optimization under uncertainties. In: Hirsch C et al (eds) *Uncertainty management for robust industrial design in aeronautics*. Springer, Cham. https://doi.org/10.1007/978-3-319-77767-2_35
- Lee SH, Choi HS, Kwak BM (2008) Multilevel design of experiments for statistical moment and probability calculation. *Struct Multidiscip O* 37(1):57–70. <https://doi.org/10.1007/s00158-007-0215-2>
- Lee SH, Chen W, Kwak BM (2009) Robust design with arbitrary distributions using Gauss-type quadrature formula. *Struct Multidiscip O* 39(39):227–243. <https://doi.org/10.1007/s00158-008-0328-2>
- Lee I, Choi KK, Zhao L (2011a) Sampling-based RBDO using the stochastic sensitivity analysis and dynamic kriging method. *Struct Multidiscip O* 44(3):299–317. <https://doi.org/10.1007/s00158-011-0659-2>
- Lee I, Choi KK, Noh Y et al (2011b) Sampling-based stochastic sensitivity analysis using score functions for RBDO problems with correlated random variables. *ASME J Mech, Des* 133(2):021003. <https://doi.org/10.1115/1.4003186>
- Li J, Gao ZH, Huang JY, Zhao K (2013) Robust design of NLF airfoils. *Chin J Aeronaut* 26(2):309–318. <https://doi.org/10.1016/j.cja.2013.02.007>
- Lucor D, Karniadakis GE (2004) Adaptive generalized polynomial chaos for nonlinear random oscillators. *SIAM J Sci Comput* 26(2):720–735. <https://doi.org/10.1137/S1064827503427984>
- Miranda J, Kumar D, Lacor C (2016) Adjoint-based robust optimization using polynomial chaos expansions. In: VII European congress on computational methods in applied sciences and engineering, 5–10 June, Crete Island, Greece
- Perkó Z, Gilli L, Lathouwers D, Kloosterman JL (2014) Grid and basis adaptive polynomial chaos techniques for sensitivity and uncertainty analysis. *J Comput Phys* 260(3):54–84. <https://doi.org/10.1016/j.jcp.2013.12.025>
- Rehman S, Matthijs L, Fred VK (2014) Efficient kriging-based robust optimization of unconstrained problems. *J Comput Phys* 5(6):872–881. <https://doi.org/10.1016/j.jocs.2014.04.005>
- Rumpfkeil MP (2013) Robust design under mixed aleatory/epistemic uncertainties using gradients and surrogates. *J Uncertainty Anal Appl* (1):7. <https://doi.org/10.1186/2195-5468-1-7>
- Suryawanshi A, Ghosh D (2016) Reliability based optimization in aero-elastic stability problems using polynomial chaos based metamodels. *Struct Multidiscip O* 53:1069–1080. <https://doi.org/10.1007/s00158-015-1322-0>
- Wang GL, Liu PQ (2016) A method of shape parameterization based on B-spline for wing design. *Civil Aircr Desig Res* (3):6–15. (In Chinese). <https://doi.org/10.19416/j.cnki.1674-9804.2016.03.002>
- Wang F, Yang S, Xiong F et al (2019a) Robust trajectory optimization using polynomial chaos and convex optimization. *Aerosp Sci Technol* 92:314–325. <https://doi.org/10.1016/j.ast.2019.06.011>
- Wang F, Xiong F, Chen S et al (2019b) Multi-fidelity uncertainty propagation using polynomial chaos and Gaussian process modeling. *Struct Multidiscip O* 4(60):1583–1604. <https://doi.org/10.1007/s00158-019-02287-7>
- Wu X, Zhang W, Song S (2017) Robust aerodynamic shape design based on an adaptive stochastic optimization framework. *Struct Multidiscip O* 57(3):1–13. <https://doi.org/10.1007/s00158-017-1766-5>
- Wu X, Zhang W, Song S (2018) Robust aerodynamic shape design based on an adaptive stochastic optimization framework. *Struct Multidiscip O* 57:639–651. <https://doi.org/10.1007/s00158-017-1766-5>
- Xia Z, Luo J, Liu F (2019) Performance impact of flow and geometric variations for a turbine blade using an adaptive NIPC method. *Aerosp Sci Technol* 90:127–139. <https://doi.org/10.1016/j.ast.2019.04.025>
- Xiong FF, Xiong Y, Greene S et al (2010) A new sparse grid based method for uncertainty propagation. *Struct Multidiscip O* 27(3):1118–1139. <https://doi.org/10.1007/s00158-009-0441-x>
- Xiong FF, Yang SX, LIU Y et al (2015a) Analysis method of engineering probability uncertainty. Science Press (In Chinese), Beijing. <https://doi.org/10.1016/j.tws.2007.05.007>
- Xiong F, Xiong Y, Xue B (2015b) Trajectory optimization under uncertainty based on polynomial chaos expansion. AIAA Guidance, Navigation, and Control Conference, 5–9 Jan, Kissimmee, Florida. AIAA 2015-1761. <https://doi.org/10.2514/6.2015-1761>
- Xiu D, Karniadakis GE (2002) The Wiener-Askey polynomial chaos for stochastic differential equations. *SIAM J Sci Comput* 24(2):619–644. <https://doi.org/10.1137/S1064827501387826>
- Xiu D, Karniadakis GE (2003) Modeling uncertainty in flow simulations via generalized polynomial chaos. *J Comput Phys* 187:137–167. [https://doi.org/10.1016/S0021-9991\(03\)00092-5](https://doi.org/10.1016/S0021-9991(03)00092-5)
- Zhang Y (2013) “Efficient uncertainty quantification in aerospace analysis and design.” Ph.D dissertation, Missouri University of Science and Technology
- Zhang Y, Serhat H (2013) Robust design optimization under mixed uncertainties with stochastic expansions. *J Mech Design* 135(8):081005. <https://doi.org/10.1115/1.4024230>
- Zhang Y, Zhu P, Chen GL (2007) Lightweight design of automotive front side rail based on robust optimization. *Thin Wall Struct* 45(7):670–676. <https://doi.org/10.1016/j.tws.2007.05.007>