



Sparse polynomial chaos expansion based on Bregman-iterative greedy coordinate descent for global sensitivity analysis

Jian Zhang^{a,*}, Xinxin Yue^a, Jiajia Qiu^{b,*}, Lijun Zhuo^a, Jianguo Zhu^a

^aDepartment of Mechanics and Engineering Science, Jiangsu University, Zhenjiang 212013, China

^bZhenjiang Intelligent Manufacturing Innovation Institute, 101 Nan-Xu Road, Zhenjiang 212016, China

ARTICLE INFO

Article history:

Received 27 May 2020

Received in revised form 15 January 2021

Accepted 3 February 2021

Keywords:

Sparse polynomial chaos expansion

Global sensitivity analysis

Greedy coordinate descent

Bregman iteration

ℓ_1 -minimization

Uncertainty quantification

ABSTRACT

Polynomial chaos expansion (PCE) is widely used in a variety of engineering fields for uncertainty and sensitivity analyses. The computational cost of full PCE is unaffordable due to the ‘curse of dimensionality’ of the expansion coefficients. In this paper, a novel methodology for developing sparse PCE is proposed by making use of the efficiency of greedy coordinate descent (GCD) in sparsity exploitation and the capability of Bregman iteration in accuracy enhancement. By minimizing an objective function composed of the ℓ_1 norm (sparsity) of the polynomial chaos (PC) coefficients and regularized ℓ_2 norm of the approximation fitness, the proposed algorithm screens the significant basis polynomials and builds an optimal sparse PCE with model evaluations much fewer than unknown coefficients. To validate the effectiveness of the developed algorithm, several benchmark examples are investigated for global sensitivity analysis (GSA). A detailed comparison is made with the well-established orthogonal matching pursuit (OMP), least angle regression (LAR) and two adaptive algorithms. Results show that the proposed method is superior to the benchmark methods in terms of accuracy while maintaining a better balance among accuracy, complexity and computational efficiency.

© 2021 Elsevier Ltd. All rights reserved.

1. Introduction

Along with the ever-increasing complexity of computer models for engineering simulations, the inherent uncertainties of input data and model parameters are evolving rapidly. In this context, characterizing the uncertainties within a computer model is of great importance, and this has motivated the development of a variety of numerical techniques for the emerging field of uncertainty quantification. To efficiently quantify the effect of variation in each input parameter on model outputs, one popular technique is to substitute a computationally expensive model with a surrogate model that possesses similar quantities of interest such as statistical moments and the distribution of model outputs.

Surrogate model (also known as metamodel) is a mathematical or numerical approximation of a complex model generated by mapping from a small amount of random inputs to the corresponding model outputs. Over the past few years, a number of surrogate models have been developed in the field of uncertainty quantification, for example polynomial regression model [1], radial basis function [2], Kriging [3]/Gaussian process [4], artificial neural network [5], support vector regression [6,7], ensemble of surrogates [8] and PCE [9–15], among which PCE has received much attention for uncertainty and sensitivity analyses [9–16].

* Corresponding authors.

E-mail addresses: jianzhang@u.nus.edu (J. Zhang), chewjia@163.com (J. Qiu).

First introduced by Ghanem and Spanos [17] to stochastic mechanics based on homogeneous chaos theory [18] and later generalized by Xiu and Karniadakis [19] for different types of statistical distributions (e.g. uniform, beta and gamma), the PCE approach is to represent explicitly the stochastic model response as a series of orthonormal multivariate polynomials, i.e. PC basis [17]. In this scenario, quantification of the response probability density function is equivalent to estimation of the PC coefficients that are the coordinates of the stochastic response in the basis and can be evaluated at a set of sampling points in the input space. To build a PCE, two main approaches are typically adopted: *intrusive* and *non-intrusive*. The intrusive approach requires modifying the solving scheme of the deterministic governing equations of the model [20], whereas the non-intrusive approaches such as the projection method [21] and the regression method [9,22] compute the PC coefficients by performing repeated simulations on limited number of input–output samples. Nevertheless, for either intrusive or non-intrusive approaches, the number of model evaluations (i.e. the computational cost) required for computing the PC coefficients increases dramatically with the number of input variables and the order of expanded polynomials.

To circumvent the issue of ‘curse of dimensionality’, several efficient non-intrusive approaches have been developed in recent years, such as adaptive methods [10,23–25] for sequentially selecting the significant basis polynomials from the full PCE, multi-fidelity methods [4,26] for achieving accurate predictions of quantities of interest using combination of ‘low-fidelity’ and ‘high-fidelity’ PCE simulations, sparse grid based methods [27,28] for utilizing sparse grid interpolation techniques to reduce the number of collocation points in constructing the PCE, ℓ_1 -minimization methods [26,29–31] for scanning important bases by minimizing the ℓ_1 norm of PCE coefficients while preserving the fitting accuracy, and Bayesian methods [32,33] for generating the sparse PCE with statistical model selection criteria. These approaches have been demonstrated with considerable computational gains compared to the classical PCE method.

In the field of signal processing and data analysis, coordinate descent (CD) algorithms have received much attention due to its simplicity and efficiency in sparsity exploitation [34–37]. On the other hand, the least absolute shrinkage and selection operator (LASSO) is widely adopted to perform continuous model selection and enforce sparse solutions for problems where the number of predictors exceeds the number of cases [38]. It was found by Wu and Lange [35] that both cyclic CD, also known as pathwise coordinate descent (PCD) [34], and GCD were superior to the LAR [39] in terms of efficiency, robustness and model selection for LASSO-penalized ℓ_2 regression, while GCD was substantially faster than cyclic CD for LASSO-penalized ℓ_1 regression. GCD was further developed by Li and Osher [40] with combination of Bregman iteration [41] to solve the compressed sensing problems [42–44]. Recently, Zhou et al. [45] developed an adaptive method based on partial least squares and distance correlation for building sparse PCE and compared their algorithm with LASSO-based sparse PCE, in which PCD [36] was employed.

Motivated by the preceding analysis, the present paper aims at efficiently building sparse PCEs in the context of LASSO-based regression by taking advantage of both GCD and Bregman iteration. To the best of the authors’ knowledge, there is still no research on the capability of GCD algorithms for the purpose of sparse PCE construction. It has been shown that GCD may converge to a sparse solution significantly faster than cyclic or randomized CD [35,46], especially for high-dimensional problems. In particular, GCD applied to problems of the LASSO form can sometimes approach to an optimal solution before executing even a single pass of all coordinates [46]. This suggests that GCD has an inherent screening ability for sparse optimization, which strongly motivates its combination with sparse PCE metamodeling for tackling high-dimensional problems. The novelty and contribution of the proposed method lie in the following aspects: (1) This study is probably the first work to develop a GCD algorithm for sparse PCE metamodeling in the context of LASSO-based regression. By updating the coordinate with the largest energy decrease [40], the GCD method is straightforward, efficient and robust in sparsity exploitation. (2) The regularization parameter λ in the LASSO-based regression is commonly selected by cross-validation [35], which often leads to inefficient computation. In addition, the inefficiency is created by the dilemma that bigger λ is preferred for more accuracy whereas smaller λ gives rise to faster convergence in the coordinate updating of GCD [40]. To tackle this issue, this study incorporates Bregman iteration into the GCD to form a Bregman-iterative GCD (BGCD) algorithm structure, which not only settles the above inefficiency by adopting a moderate λ with a relatively wide range of appropriate values, but also considerably enhances the convergence and accuracy of the GCD in solving the PC coefficients. (3) Existing non-intrusive algorithms for building sparse PCEs are mainly for problems with more cases than the number of predictors. However, the proposed algorithm is devised to handle problems with far fewer cases than the number of predictors, which is of great potential to the uncertainty quantification in practical engineering.

In this paper, the sparse PCE is employed for the GSA that aims at quantifying the respective effects of different inputs and their interactions on an assigned output response. GSA can provide complete information about the model behavior when the inputs vary in the entire domain. The priority level and ranking of the inputs resulting from the GSA can be very helpful for designers in narrowing the uncertain scope of model response. Among the developed works on sensitivity analysis, Sobol’ indices have attracted greater portion of attention due to the fact that they can provide accurate information for most models [16,47–49]. The remainder of the paper is organized as follows. Section 2 gives a brief introduction of the Sobol’ decomposition and corresponding sensitivity indices for the GSA. In Section 3, the polynomial chaos approximation of a multidimensional model is recalled. The detailed procedure on how to determine the Sobol’ indices from the PC coefficients is given in particular. Then, in Section 4, a BGCD algorithm is proposed for building the optimal sparse PCE from a given sample set. The performance of the proposed algorithm is assessed on several benchmark examples for GSA in Section 5, and the results are compared with the well-established OMP algorithm, LAR technique and two adaptive methods. Section 6 summarizes concluding remarks.

2. Sobol' decomposition

In this section, the Sobol' decomposition and corresponding sensitivity indices for the GSA are briefly recalled.

Let us consider a square integrable function $y = \mathcal{F}(\mathbf{x})$ having n -dimensional independent inputs defined in the unit hypercube $[0, 1]^n$. The Sobol' decomposition of $\mathcal{F}(\mathbf{x})$ into summands of increasing dimensions can be represented as follows [47–49]:

$$\mathcal{F}(x_1, \dots, x_n) = \mathcal{F}_0 + \sum_{i=1}^n \mathcal{F}_i(x_i) + \sum_{1 \leq i < j \leq n} \mathcal{F}_{ij}(x_i, x_j) + \dots + \mathcal{F}_{1,2,\dots,n}(x_1, \dots, x_n), \quad (1)$$

where \mathcal{F}_0 is a constant and the integral of each summand $\mathcal{F}_{i_1,\dots,i_s}(x_{i_1}, \dots, x_{i_s})$ over any of its independent variables is zero. The pairwise orthogonality of the summands in the decomposition (1) can be expressed in the following sense:

$$\mathbb{E}[\mathcal{F}_{i_1,\dots,i_s}(x_{i_1}, \dots, x_{i_s}) \times \mathcal{F}_{j_1,\dots,j_t}(x_{j_1}, \dots, x_{j_t})] = 0 \text{ for } \{i_1, \dots, i_s\} \neq \{j_1, \dots, j_t\}. \quad (2)$$

Integrate the square of Eq. (1) over $[0, 1]^n$ and we can obtain the following equation:

$$\int \mathcal{F}^2(\mathbf{x}) d\mathbf{x} = \mathcal{F}_0^2 + \sum_{i=1}^n \int \mathcal{F}_i^2(x_i) dx_i + \dots + \int \mathcal{F}_{1,2,\dots,n}^2(x_1, \dots, x_n) dx_1 \dots dx_n. \quad (3)$$

It can be easily deduced that

$$D = \sum_{i=1}^n D_i + \sum_{1 \leq i < j \leq n} D_{ij} + \dots + \sum_{1 \leq i_1 < \dots < i_s \leq n} D_{i_1,\dots,i_s} + \dots + D_{1,2,\dots,n}, \quad (4)$$

where D is the total variance of $\mathcal{F}(\mathbf{x})$, and D_{i_1,\dots,i_s} is the partial variance due to the interactive effect of $\{x_{i_1}, \dots, x_{i_s}\}$.

To this end, the Sobol' indices are defined in the following form:

$$S_{i_1,\dots,i_s} = \frac{D_{i_1,\dots,i_s}}{D}. \quad (5)$$

Therefore, each sensitivity index S_{i_1,\dots,i_s} measures which amount of the total variance D is attributed to the uncertainties in the input random variables $\{x_{i_1}, \dots, x_{i_s}\}$. The first-order sensitivity index S_i represents the influence due to x_i alone while the higher order index accounts for the cooperative influence of various variables. To evaluate the total effect of an input variable x_i on the variance of y , the total sensitivity index S_i^T is introduced as the sum of all partial sensitivity indices S_{i_1,\dots,i_s} involving parameter i [49]:

$$S_i^T = \sum_{\zeta_i} S_{\zeta_i}, \quad \zeta_i = \{(i_1, \dots, i_s) : \exists k, 1 \leq k \leq s, i_k = i\}. \quad (6)$$

The Sobol' indices are usually computed by using Monte Carlo (MC) simulation [47,49,50]. However, a very large number of model evaluations (e.g. 2^n MC integrals needed for n input variables) are usually required to obtain accurate estimates, which is obviously infeasible for a computationally demanding model. In this regard, represented in an orthonormal polynomial basis, PCE can be used as a model substitute to reduce computational cost while keeping the prediction accuracy. In addition, due to the nature of the PC basis, the sensitivity indices can be evaluated simply as analytical functions of the PC coefficients [9].

3. Polynomial chaos approximation of the model response

3.1. Full PCE

The classic PCE of the model response $y = \mathcal{F}(\mathbf{x})$ can be represented as follows:

$$y = \mathcal{F}(\mathbf{x}) = \sum_{\alpha \in \mathbb{N}^n} \beta_{\alpha} \psi_{\alpha}(\mathbf{x}), \quad (7)$$

where $\alpha = (\alpha_1, \dots, \alpha_n)$ (with $\alpha_i \geq 0$) is an n -dimensional index, and β_{α} 's are unknown deterministic PC coefficients. The multivariate polynomial ψ_{α} is the tensor product of normalized univariate orthogonal polynomials:

$$\psi_{\alpha}(\mathbf{x}) = \prod_{i=1}^n \psi_{\alpha_i}^{(i)}(x_i). \quad (8)$$

Different types of univariate orthogonal polynomials commonly used for constructing PC are listed in Table 1 [19].

In practice, the PCE in Eq. (7) is usually truncated for computational purposes. A common way is to retain those polynomials whose total degree $|\alpha| = \sum_{i=1}^n \alpha_i$ does not exceed a given degree of p :

Table 1
Common orthogonal polynomials and their associated random variables.

Random variable	Polynomial	Support
Gaussian	Hermite	$(-\infty, \infty)$
Uniform	Legendre	$[a, b]$
Beta	Jacobi	$[a, b]$
Gamma	Laguerre	$[0, \infty)$
Poisson	Charlier	$\{0, 1, 2, \dots\}$
Binomial	Krawtchouk	$\{0, 1, 2, \dots, n\}$
Negative binomial	Meixner	$\{0, 1, 2, \dots\}$
Hypergeometric	Hahn	$\{0, 1, 2, \dots, n\}$

$$y \simeq \mathcal{F}_p(\mathbf{x}) = \sum_{\alpha \in \mathcal{A}^{p,n}} \beta_{\alpha} \psi_{\alpha}(\mathbf{x}), \mathcal{A}^{p,n} = \{\alpha \in \mathbb{N}^n : |\alpha| \leq p\}. \quad (9)$$

The expression of Eq. (9) is called the full PCE of degree p of the model response y . The total number of unknown PC coefficients P can be calculated from the maximum degree p and the dimensionality n of inputs as follows:

$$P = \binom{n+p}{p} = \frac{(n+p)!}{n!p!}. \quad (10)$$

With such a truncation, the problem of characterizing the model response y is converted into computing a finite set of unknown PC coefficients. This can be achieved using non-intrusive techniques, as described in the sequel.

3.2. Computation of the PC coefficients

In order to compute the PC coefficients, regression-based methods usually seek a PCE that satisfies:

$$\mathcal{F}_p(x_j) = \sum_{\alpha \in \mathcal{A}^{p,n}} \beta_{\alpha} \psi_{\alpha}(x_j) \approx \mathcal{F}(x_j) = y_j, \quad j = 1, \dots, N, \quad (11)$$

where N is the number of input–output samples. Eq. (11) can be rewritten in the form of $\mathbf{y} = \boldsymbol{\psi}\boldsymbol{\beta}$, where $\boldsymbol{\beta} \in \mathbb{R}^P$ is a vector of unknown PC coefficients, $\mathbf{y} \in \mathbb{R}^N$ is a vector of N realizations of the model output, and $\boldsymbol{\psi} \in \mathbb{R}^{N \times P}$ is the measurement matrix of which each column contains evaluations of the PC basis polynomials at the N samples. The PC coefficients are evaluated by minimizing the residual between the model responses and the PCE approximation. For $N \geq P$, the unknown coefficients can be computed using the least-squares regression: $\boldsymbol{\beta} = (\boldsymbol{\psi}^T \boldsymbol{\psi})^{-1} \boldsymbol{\psi}^T \mathbf{y}$. When $N < P$, the system equation for $\boldsymbol{\beta}$ becomes ill-posed, and the least-squares approach is no longer feasible. To this end, some form of regularization is usually introduced to identify a unique solution.

For a PCE with sufficient sparsity, the unknown PC coefficients in Eq. (11) in the case of $N < P$ can be determined with only a few terms of significant nonzeros by solving the following ℓ_1 -minimization problem

$$\min_{\boldsymbol{\beta} \in \mathbb{R}^P} \|\boldsymbol{\beta}\|_1 \quad \text{subject to} \quad \|\boldsymbol{\psi}\boldsymbol{\beta} - \mathbf{y}\|_2 \leq \epsilon \quad (12)$$

with an ℓ_2 norm constraint to account for the error ϵ in the p -th degree truncation of the PCE. The constrained ℓ_1 -minimization problem in Eq. (12) can be reformulated as a regularized (unconstrained) optimization problem:

$$\min_{\boldsymbol{\beta} \in \mathbb{R}^P} \mathcal{E}(\boldsymbol{\beta}) = \min_{\boldsymbol{\beta} \in \mathbb{R}^P} \left\{ \|\boldsymbol{\beta}\|_1 + \lambda \|\boldsymbol{\psi}\boldsymbol{\beta} - \mathbf{y}\|_2^2 \right\}, \quad (13)$$

where the first term of the objective (or energy) function $\mathcal{E}(\boldsymbol{\beta})$ gives the ℓ_1 norm (sparsity) of the PC coefficients, while the second term measures how well the approximation fits the true model response in the sense of ℓ_2 norm and needs to be penalized heavily for more accurate solutions. The so-called regularization parameter λ controls the relative weight of the two terms. Eq. (13) is known as the LASSO-based regression problem [35,40] or general ℓ_1 -minimization problem [41], which can be solved by two main categories of approaches: basis pursuit [51] and greedy algorithms [43,52]. In the present paper, GCD based algorithm is proposed in Section 4 to solve the above Eq. (13) for building a sparse PCE, and compared with two well-established greedy methods: OMP [43,52] and LAR [24].

3.3. Definition of sparse PCE

Let \mathcal{A} be a non-empty finite subset of \mathbb{N}^n , and the truncated PCE can be defined by:

$$\mathcal{F}_{\mathcal{A}}(\mathbf{x}) = \sum_{\alpha \in \mathcal{A}} \beta_{\alpha} \psi_{\alpha}(\mathbf{x}). \quad (14)$$

In the sequel, set \mathcal{A} is called the truncation set. The common truncation scheme in Eq. (9) corresponds to the choice $\mathcal{A} = \mathcal{A}^{p,n}$, which is called the full PCE of the model response. Since the large cardinality of this set may make the computational cost unaffordable, the determination of truncation set \mathcal{A} of small cardinality is of interest. This allows one to define the index of sparsity of \mathcal{A} by:

$$IS = \frac{\text{card}(\mathcal{A})}{\text{card}(\mathcal{A}^{p_{\max},n})}, p_{\max} = \max_{\alpha \in \mathcal{A}} (|\alpha|), \quad (15)$$

where $\text{card}(\mathcal{A})$ denotes the number of elements in \mathcal{A} , p_{\max} corresponds to the degree of the truncated PCE in Eq. (14). A PCE in (14) is said to be sparse if the index of sparsity IS is small enough compared to 1. The benefits of a sparse PCE are two-fold [46]: (1) to improve the convergence rate for GCD and (2) to allow for sparse data structures to reduce memory requirement and faster matrix–vector multiplications. Moreover, the degree and interaction order of any index α in \mathcal{A} are respectively defined by:

$$p_{\alpha} = |\alpha| = \sum_{i=1}^n \alpha_i, \quad \eta_{\alpha} = \sum_{i=1}^n \alpha_{i>0}, \quad (16)$$

where $\alpha_{i>0} = 1$ if $\alpha_i > 0$ and 0 otherwise.

3.4. PC-based global sensitivity indices

Let us define a subset of multidimensional indices $\mathcal{L}_{i_1, \dots, i_s}$ in the following sense:

$$\mathcal{L}_{i_1, \dots, i_s} = \left\{ \alpha \in \mathcal{A} : \begin{array}{ll} \alpha_k > 0 & k \in (i_1, \dots, i_s), \forall k = 1, \dots, n \\ \alpha_k = 0 & k \notin (i_1, \dots, i_s), \forall k = 1, \dots, n \end{array} \right\}. \quad (17)$$

Then, the sparse PCE in Eq. (14) can be rewritten according to Sobol' decomposition:

$$\mathcal{F}_{\mathcal{A}}(\mathbf{x}) = \beta_0 + \sum_{i=1}^n \sum_{\alpha \in \mathcal{L}_i} \beta_{\alpha} \psi_{\alpha}(x_i) + \sum_{1 \leq i_1 < i_2 \leq n} \sum_{\alpha \in \mathcal{L}_{i_1, i_2}} \beta_{\alpha} \psi_{\alpha}(x_{i_1}, x_{i_2}) + \dots + \sum_{1 \leq i_1 < \dots < i_s \leq n} \sum_{\alpha \in \mathcal{L}_{i_1, \dots, i_s}} \beta_{\alpha} \psi_{\alpha}(x_{i_1}, \dots, x_{i_s}) + \dots + \sum_{\alpha \in \mathcal{L}_{1, \dots, n}} \beta_{\alpha} \psi_{\alpha}(\mathbf{x}) \quad (18)$$

where each summand in Eq. (1) can be identified as follows: $\mathcal{F}_{i_1, \dots, i_s}(x_{i_1}, \dots, x_{i_s}) = \sum_{\alpha \in \mathcal{L}_{i_1, \dots, i_s}} \beta_{\alpha} \psi_{\alpha}(x_{i_1}, \dots, x_{i_s})$.

Due to the orthonormality of the basis polynomials, it can be easily derived from the sparse PCE representation Eq. (18) that the total and partial variances are respectively:

$$D^A = \sum_{\alpha \in \mathcal{A} \setminus \{\mathbf{0}\}} \beta_{\alpha}^2, \quad D_{i_1, \dots, i_s}^A = \sum_{\alpha \in \mathcal{L}_{i_1, \dots, i_s}} \beta_{\alpha}^2. \quad (19)$$

Then, the PC-based Sobol' indices S_{i_1, \dots, i_s}^A and total sensitivity indices $S_i^{T,A}$ obtained from the above equations are respectively:

$$S_{i_1, \dots, i_s}^A = \frac{D_{i_1, \dots, i_s}^A}{D^A}, \quad S_i^{T,A} = \sum_{\alpha: \alpha_i > 0} S_{\alpha}^A. \quad (20)$$

Once the sparse PCE of the model response is built, the Sobol' indices can be obtained analytically at a negligible computational cost for GSA. To this end, a novel BGCD algorithm to efficiently build a sparse PCE is proposed in the next section.

4. Bregman-iterative GCD for sparse PCE

To build a sparse PCE by solving the above LASSO-based optimization problem raises two concerns: one is to develop an effective algorithm for minimizing the objective function $\mathcal{E}(\beta)$, and the other is how to determine the regularization parameter λ . In this section, a novel algorithm based on the integration of GCD and Bregman iteration is developed for sparse PCE construction. First, GCD is newly developed to solve the LASSO-based regression problem (13) for building sparse PCE. Second, Bregman iteration is incorporated into the GCD to form a novel BGCD algorithm, which on one hand solves the inefficiency in determining the regularization parameter, and on the other hand improves the accuracy and convergence of the GCD. Finally, the detailed procedures of proposed algorithm are provided.

4.1. Constructing sparse PCE with GCD

In multivariable minimization, CD algorithms minimize the objective by successively solving scalar minimization subproblems along all coordinates that correspond to the PC basis polynomials $\psi_{\alpha}(\mathbf{x})$ in this study. In this regard, CD is attractive due to the fact that scalar minimization is simpler than multivariable minimization, and efficient when the subproblems can

be solved quickly. In the PCD [34,36], the coordinates are visited in a sequential order to reach a minimum. For some problems, however, the order in which the coordinates are visited may have a significant effect on the rate of convergence. Therefore, a GCD with an adaptive order [40] that converges faster than the PCD is employed in this study.

The objective function $\mathcal{E}(\boldsymbol{\beta})$ in the LASSO-based regression problem (13) has the additive decomposition in the form of

$$\mathcal{E}(\beta_1, \dots, \beta_p) = g(\beta_1, \dots, \beta_p) + \sum_{j=1}^p h_j(\beta_j), \quad (21)$$

where $g(\boldsymbol{\beta}) = \lambda \|\boldsymbol{\psi}\boldsymbol{\beta} - \mathbf{y}\|_2^2 : \mathbb{R}^p \rightarrow \mathbb{R}$ is differentiable and convex, and the univariate functions $h_j(\beta_j) = |\beta_j| : \mathbb{R} \rightarrow \mathbb{R}$ are convex (but not necessarily differentiable). It was shown by Tseng [53] that for any convex objective function \mathcal{E} with the separable structure (21), the CD algorithm is guaranteed to converge to the global minimizer. The key property underlying this result is the separability of the nondifferentiable component $h(\boldsymbol{\beta}) = \sum_{j=1}^p h_j(\beta_j)$, as a sum of functions of each individual parameter. This ensures the minimization problem (13) to be solved by CD algorithm through a sequence of one-dimensional optimizations.

For each coordinate subproblem, all components of $\boldsymbol{\beta}$ except the j -th component β_j are fixed. Let ϕ_j denote the j -th column of $\boldsymbol{\psi}$, ϕ_{ij} the element of $\boldsymbol{\psi}$ in the i -th row and j -th column, and y_i the i -th component of \mathbf{y} , the problem is to minimize:

$$\min_{\beta_j} \left\{ |\beta_j| + \sum_{k \neq j} |\beta_k| + \lambda \sum_{i=1}^N \left(\phi_{ij}\beta_j + \sum_{k \neq j} \phi_{ik}\beta_k - y_i \right)^2 \right\}. \quad (22)$$

From the theory of convex analysis, it is known that necessary and sufficient conditions for a solution to problem (22) take the form [54]

$$s_j + 2\lambda \sum_{i=1}^N (\phi_{ij}\beta_j - y_i)\phi_{ij} = 0, \quad (23)$$

where s_j is a subgradient of $|\beta_j|$ and equal to $\text{sign}(\beta_j)$ if $\beta_j \neq 0$ and some value lying in $[-1, 1]$ otherwise. It is seen from Eq. (22) that solution for each β_j can be expressed succinctly in terms of the partial residual $r_j = y_i - \sum_{k \neq j} \phi_{ik}\beta_k$, which removes from the outcome the current fit from all but β_j . As a result, β_j is updated as

$$\tilde{\beta}_j = \frac{1}{\|\phi_j\|_2^2} \text{shrink}\left(\gamma_j, \frac{1}{2\lambda}\right), \quad (24)$$

where $\|\phi_j\|_2^2 \neq 0$ is assumed; otherwise, $\phi_{ij} = 0$ for any i , and the whole j -th column of $\boldsymbol{\psi}$ can be deleted. The shrink operator (i.e. soft-thresholding operator [54]) is defined as:

$$\text{shrink}(y, \mu) = \text{sgn}(y) \max\{|y| - \mu, 0\} = \begin{cases} y - \mu, & \text{if } y \in (\mu, \infty); \\ 0, & \text{if } y \in [-\mu, \mu]; \\ y + \mu, & \text{if } y \in (-\infty, -\mu) \end{cases} \quad (25)$$

with $\gamma_j = \sum_i \phi_{ij}(y_i - \sum_{k \neq j} \phi_{ik}\beta_k)$. The overall algorithm operates by applying this shrink update (24) repeatedly in some fixed manner, updating the coordinates of $\tilde{\boldsymbol{\beta}}$ (and hence the residual vector) along the way.

To build the sparse PCE efficiently, in this paper the coordinate that produces the largest decrease in the energy function is selected:

$$j^* = \arg \max_j \Delta \mathcal{E}_j. \quad (26)$$

where $\Delta \mathcal{E}_j = \mathcal{E}(\beta_1, \dots, \beta_j, \dots, \beta_p) - \mathcal{E}(\beta_1, \dots, \tilde{\beta}_j, \dots, \beta_p)$. Thus, computational savings are obtained by updating γ_j instead of entirely recomputing it through every iteration. The following expression for γ_j can be obtained:

$$\gamma_j^{k+1} - \gamma_j^k = \phi_j^T \boldsymbol{\psi} (\boldsymbol{\beta}^k - \boldsymbol{\beta}^{k+1}) + \|\phi_j\|_2^2 (\beta_j^{k+1} - \beta_j^k). \quad (27)$$

Suppose the q -th coordinate is chosen to be updated in the k -th iteration, then $\boldsymbol{\beta}^{k+1} - \boldsymbol{\beta}^k$ is non-zero only in the q -th coordinate. So, $\boldsymbol{\beta}^{k+1} - \boldsymbol{\beta}^k = (\beta_q^{k+1} - \beta_q^k) \mathbf{e}_q$, where \mathbf{e}_q is the q -th standard basis vector. Eq. (27) becomes

$$\gamma_j^{k+1} - \gamma_j^k = (\beta_q^{k+1} - \beta_q^k) (\|\phi_q\|_2^2 \mathbf{I} - \boldsymbol{\psi}^T \boldsymbol{\psi}) \mathbf{e}_q. \quad (28)$$

The above steps of Eqs. (24)–(28) constitute the GCD algorithm for solving the minimization problem Eq. (13) as in the sequel.

4.2. Modification of GCD-based sparse PCE with Bregman iteration

In the preceding algorithm, the value of regularization parameter λ affects the accuracy of the obtained PC coefficients. On one hand, large λ is preferred to heavily penalize the approximation fitness of ℓ_2 norm in Eq. (13) for more accurate solutions. On the other hand, if λ is of big value, the threshold of $\frac{1}{\lambda}$ is very small which consequently leads to slow convergence by using the GCD algorithm. To tackle this situation, Bregman iteration [41] is adopted in combination with the above GCD to conquer the inefficiency issue of λ .

Bregman distance [55] based on a convex functional $J(\cdot)$ between points β and \mathbf{v} is

$$B_j^c(\mathbf{v}, \beta) = J(\mathbf{v}) - J(\beta) - \langle \mathbf{v} - \beta, c \rangle, \tag{29}$$

where $c \in \partial J(\beta)$ is an element in the subgradient of J at the point β . Here, $J(\beta) = \|\beta\|_1$, and the convex and differentiable function $H(\beta, \mathbf{y}) = \lambda \|\psi\beta - \mathbf{y}\|_2^2$. After the PC coefficients β^{k+1} are iteratively solved by using the GCD procedures as in Section 4.1:

$$\beta^{k+1} \leftarrow \arg \min_{\beta} J(\beta) + \lambda \|\psi\beta - \mathbf{y}^k\|_2^2, \tag{30}$$

Bregman iteration is introduced as follows:

$$\mathbf{y}^{k+1} \leftarrow \mathbf{y} + (\mathbf{y}^k - \psi\beta^{k+1}). \tag{31}$$

The advantages of using Bregman iteration are two-fold: (1) for any λ in the subproblem (see Eq. (30)), Bregman iteration will converge to the exact solution of the constrained problem Eq. (12). In this regard, λ should be selected such that the subproblem can be solved efficiently. As stated before, a very big λ results in slow convergence. In the other extreme, when λ is too small, more Bregman iterations are needed. So the optimal value for λ is something intermediate. Numerical experiments show that, by choosing an appropriate λ , both the subproblems can be solved quickly and few Bregman iterations are needed; (2) due to the efficiency of Bregman iteration, it is not necessary to solve the subproblem (see Eq. (30)) very accurately, which is computationally demanding. In other words, with the help of few Bregman iterations, the solution accuracy can be enhanced on top of solving the LASSO-based regression problem Eq. (13) by the GCD.

4.3. The proposed algorithm

The flowchart of the BGCD for sparse PCE is illustrated in Fig. 1. It has two layers of loops: (1) the inner loop uses GCD to solve the subproblems of Eq. (13) always along the coordinate producing the largest decrease in the energy function $\mathcal{E}(\beta)$, and obtain the optimal value of β through iterating γ ; (2) the outer loop adopts Bregman iteration for fast convergence and accuracy enhancement of the PC coefficients through updating the response value \mathbf{y} .

Suppose an experimental design $\mathbf{x} = (\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(N)})$ with N realizations is generated. The Sobol' quasi-random sequence [56], also known as low discrepancy sequences, is adopted for generating samples in this research due to its space filling property. After running the model at the design points, the corresponding model responses of interest are gathered into the vector $\mathbf{y} = (y^{(1)}, y^{(2)}, \dots, y^{(N)})$. The proposed algorithm is outlined as follows:

Step 1 (*Initialization*): Define the total degree p and interaction order η of the full PCE. Set $P = \text{card}(\mathcal{A}^{p,\eta})$ and define the basis polynomial vector $\psi = (\psi_1, \psi_2, \dots, \psi_p)$ associated to $\mathcal{A}^{p,\eta}$. Since the BGCD algorithm is devised for underdetermined case with fewer equations than unknown PC coefficients, the basis polynomial order should be selected with the requirement of $P > N$. Then, the following initial vectors are defined: $\gamma^1 = \psi^T \mathbf{y}$, $\beta^0 = \mathbf{0}$, and $\mathbf{y}^0 = \mathbf{0}$.

Step 2 (*Calculating the optimal value for β*): Define the regularization parameter $\lambda = p \times n + 2$, calculate $\mathbf{w} = \text{diag}(\psi^T \psi)$ and obtain the optimal value $\tilde{\beta}^k = \text{shrink}\left(\frac{\gamma^k}{\mathbf{w}}, \frac{1}{2\lambda \mathbf{w}}\right)$ in the k -th iteration with Eq. (24).

Step 3 (*Comparison of the iterated PC coefficients*): Calculate the differences between the PC coefficients obtained in the k -th and $(k-1)$ -th iterations, and compare $\sum_i \Delta\beta_i$ with a given threshold value To1 . As Bregman iteration will be used in the outer loop, the subproblem in Eq. (30) is not necessary to be solved very accurately. In this study, the threshold value used for To1 ranges from 10^{-6} to 10^{-4} .

Step 4 (*Updating the coordinate with the largest energy decrease*): If $\sum_i \Delta\beta_i > \text{To1}$, choose the coordinate which produces the largest decrease in the energy function with Eq. (26). Then, according to Eq. (28), update $\beta_i^{k+1} = \beta_i^k$ for $i \neq j$, $\beta_j^{k+1} = \tilde{\beta}_j^k$, $\gamma^{k+1} = \gamma^k + (\beta_j^k - \tilde{\beta}_j^k)(\psi^T \psi - w_j \mathbf{I}) e_j$ and $\gamma_j^{k+1} = \gamma_j^k$. Set $k = k + 1$, and resume from Step 2. When $\sum_i \Delta\beta_i \leq \text{To1}$, continue Step 5 of the algorithm.

Step 5 (*Bregman iteration for updating \mathbf{y}*): If $\frac{\|\psi\beta^k - \mathbf{y}\|_2}{\|\mathbf{y}\|_2} > \text{To2}$, update $\mathbf{y}^{k+1} = \mathbf{y} + (\mathbf{y}^k - \psi\beta^{k+1})$. Set $k = k + 1$ and go back to Step 2. When $\frac{\|\psi\beta^k - \mathbf{y}\|_2}{\|\mathbf{y}\|_2} \leq \text{To2}$, the stopping criterion is satisfied, and the optimal value of PC coefficients is obtained. As Bregman iteration can enhance the accuracy in solving the general ℓ_1 -minimization problem by using GCD within few iterations, the threshold value of To2 can be much larger compared to that of To1 . In this study, To2 from 10^{-3} to 10^{-2} is adopted.

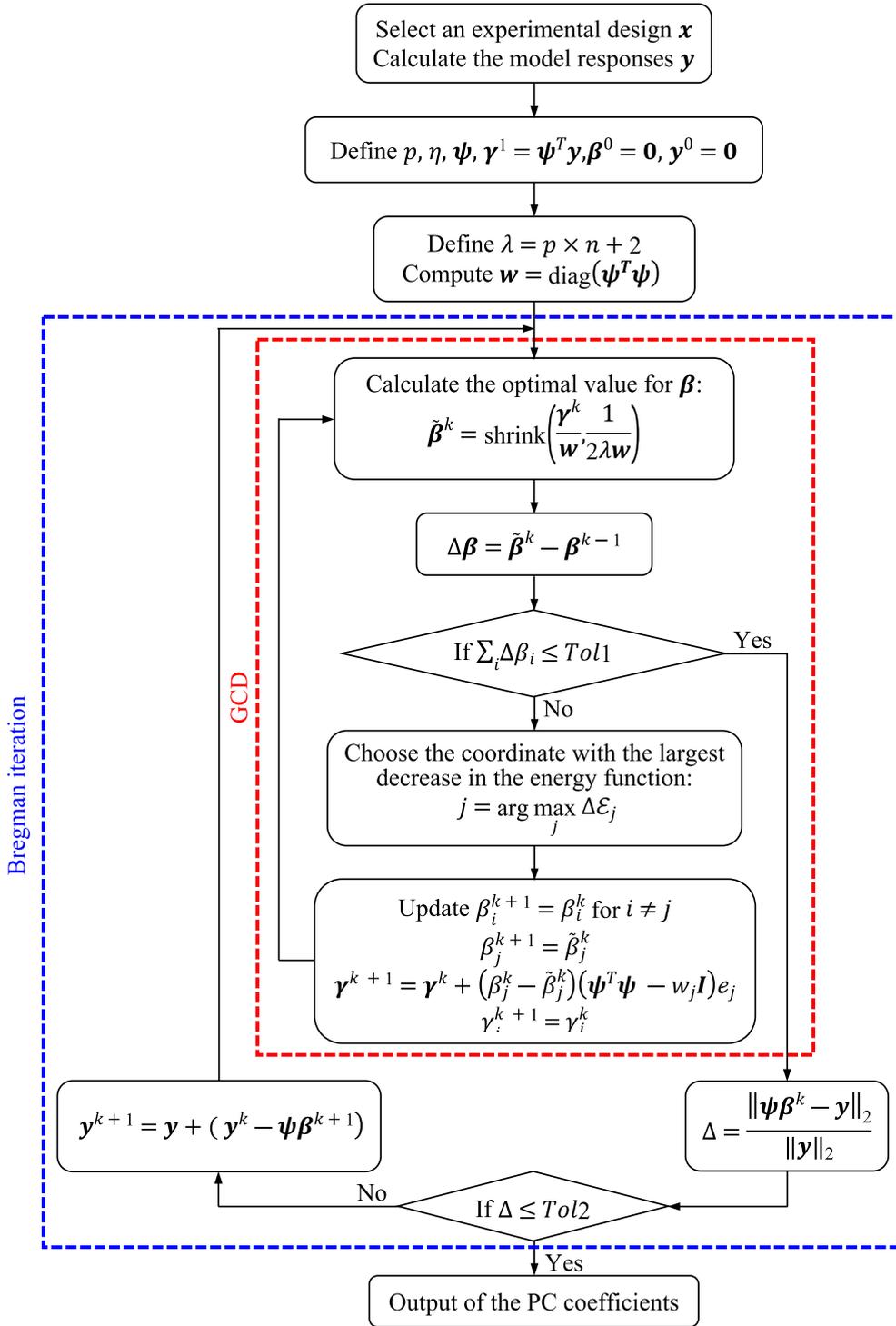


Fig. 1. Flowchart of the BGCD for sparse PCE.

GCD consists of Step 2 to Step 4, and Bregman iteration is adopted in Step 5. Thanks to Bregman iteration, an intermediate value of λ can be selected so that the coordinate subproblems can be solved quickly by the GCD, and as few as one to eight Bregman iterations are needed to get converged solutions, as demonstrated in the following section. In this work, $\lambda = p \times n + 2$ is used according to the problem dimensionality n and the PCE degree p for all numerical examples in Section 5.

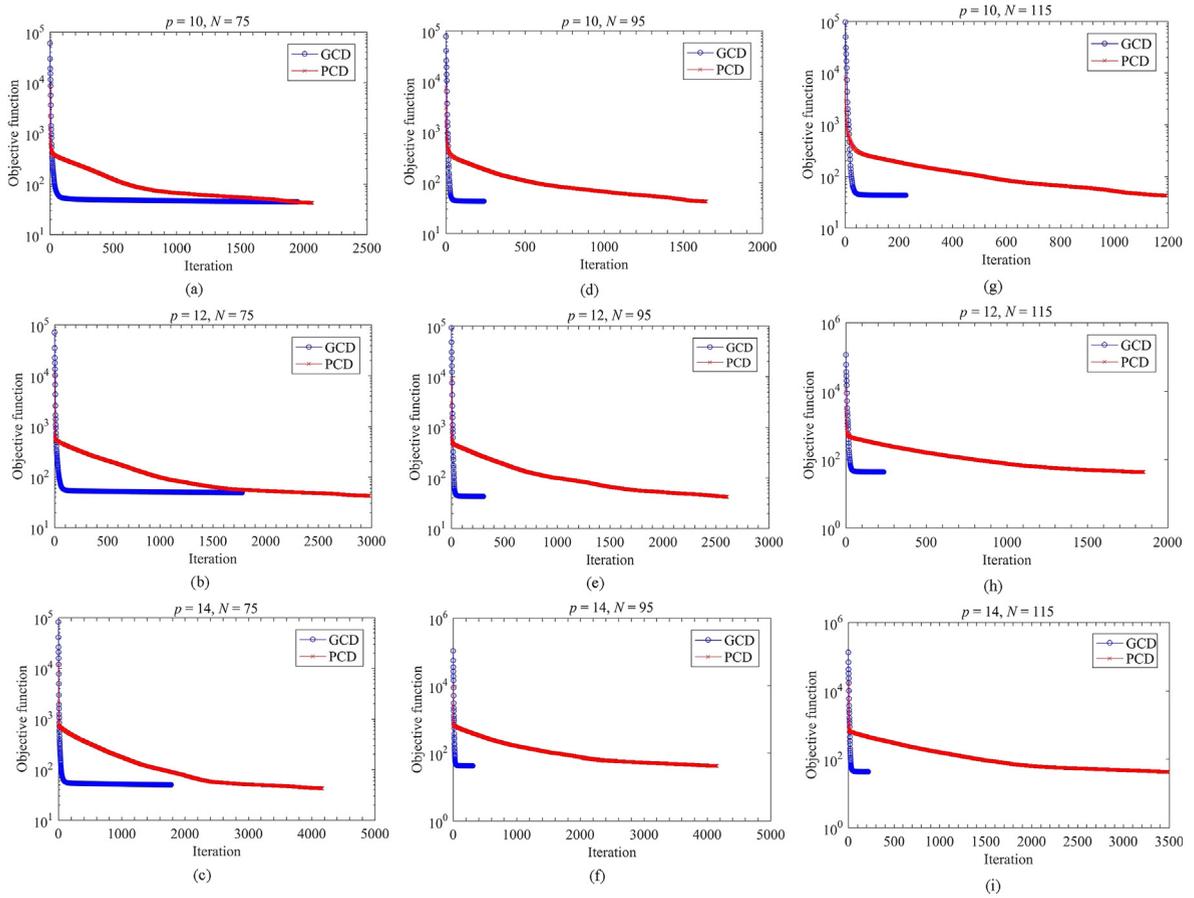


Fig. 2. Ishigami function–Convergence of the objective function with different degrees of the PCE and samples: $p = 10, 12$ and 14 , and $N = 75, 95$ and 115 .

5. Numerical examples

This section is devoted to the validation and evaluation of the proposed algorithm for building sparse PCE. Three benchmark functions are first considered: the Ishigami function, Sobol’ function and Morris function. The overall performances of the proposed BGCD algorithm are compared with the PCD, GCD, OMP, LAR and two adaptive algorithms [45,10]. Both OMP and LAR are with a MATLAB implementation called SparseLab available at <http://sparselab.stanford.edu/>. The quality of the sparse PCE model is tested with different samples and degrees of the PCE. The estimated sensitivity indices are compared to analytical values by evaluating the following sensitivity index error:

$$e_S = \sum_{i=1}^n \left| S_i^A - S_i^{exact} \right| + \sum_{i=1}^n \left| S_i^{T,A} - S_i^{T,exact} \right|, \tag{32}$$

where the superscript “exact” stands for the analytical value. The good performance of the developed algorithm is eventually demonstrated on two engineering examples of finite element analysis, consisting of a planar and a spatial truss structure with 9 and 21 input random variables respectively.

5.1. Example 1: Ishigami function

Let us consider the Ishigami function of high nonlinearity and non-monotonicity, which is widely used for benchmarking in GSA [57]:

$$Y = \sin X_1 + a \sin^2 X_2 + b X_3^4 \sin X_1, \tag{33}$$

where the input variables $X_i (i = 1, 2, 3)$ are uniformly distributed over $[-\pi, \pi]$, $a = 7$ and $b = 0.1$. Note that this function is sparse in nature since: (1) the model has three independent variables but the maximum interaction order is 2; (2) the function is of even order with respect to the variables X_2 and X_3 , hence the odd polynomials of these variables are zero in the PCE.

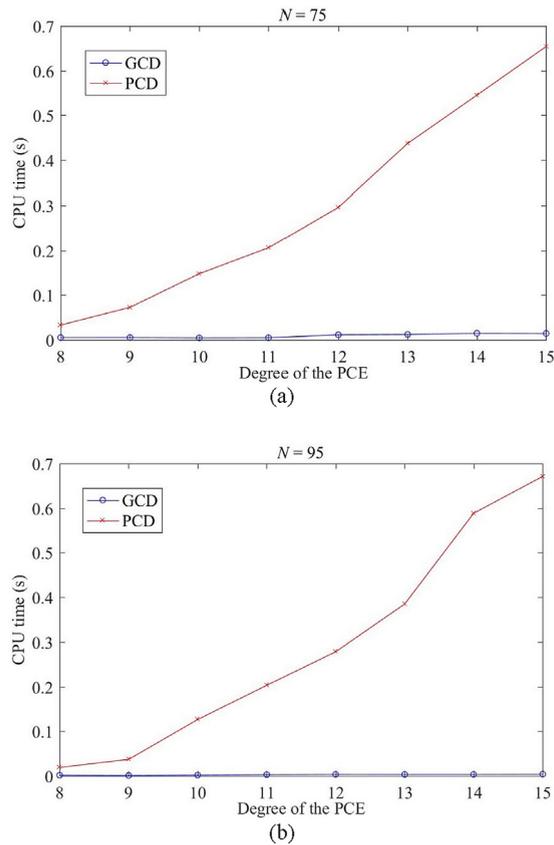


Fig. 3. Ishigami function–CPU time (s) for (a) 75 and (b) 95 samples with different degrees of the PCE.

The sensitivity indices of the model response (33) can be calculated analytically as in [16]. Here, they are approximated by postprocessing a sparse PCE of the model response according to Eq. (20). The sparse PC coefficients are evaluated using Sobol’ quasi-random sequences of different sizes $N = 55, 65, \dots, 115$. The degree p of the PCE varies from 8 to 15.

First, the convergence and computational efficiency of the GCD (without Bregman iteration) and PCD are compared. Fig. 2 represents the value of the objective function varying with iterations using different degrees of the PCE and samples. The algorithm terminates if the difference of the objective function values between two consecutive iterations is less than 10^{-3} . It is observed that when N is fixed at 75 with p varying from 10 to 14 (see Fig. 2(a)–(c)), the number of iteration for PCD to converge increases rapidly from 2068 to 4166 whereas that for GCD to converge changes slightly from 1953 to 1783. Similar observations are obtained for respective $N = 95$ and $N = 115$ with p varying from 10 to 14 (see Fig. 2(d)–(i)): the numbers of iteration for PCD to converge increase respectively from 1645 to 4156 and from 1193 to 3490, whereas those for GCD to converge vary slightly from 242 to 311 and from 227 to 219 respectively. This indicates that the convergence velocity of the objective function by using PCD is negatively related to the degree of the PCE, whereas that by using GCD does not change much with respect to p . On the other hand, it is found that when p is respectively fixed at 10, 12 and 14 with N varying from 75 to 115, the convergence velocity of the objective function by using either PCD or GCD is positively related to the sample number. It is worth noting that when $N = 75$, the convergence velocity of the objective function by using GCD is at a similar level to that by using PCD. However, the convergence velocity of the objective function by using GCD is 6 to 12 times faster than that by using PCD when N is increased to 95, and 4 to 15 times faster than that by using PCD if N is further increased to 115. Comparisons of the CPU time (with an AMD 3.60 GHz CPU and 16 GB RAM, the same below) for $N = 75$ and 95 with different degrees of the PCE and for $p = 10$ and 12 with different samples by using GCD and PCD are shown in Fig. 3 and Fig. 4, respectively. As the degree of the PCE or sample varies, the CPU time by using PCD changes more significantly than that by using GCD. In this example, GCD is 10 to 160 times more efficient than PCD for the considered p and N . Considering the results of both the convergence velocity of the objective function in Fig. 2 and CPU time in Figs. 3 and 4, $N = 95$ and $p = 10$ are recommended for the GCD in this example. For other examples in this paper, similar strategy can be applied to determine the optimal values of N and p .

Second, effects of Bregman iteration and two threshold values ($To1$ and $To2$) on the accuracy and efficiency of the GCD are investigated. Table 2 lists the results of sensitivity index error e_s and CPU time varying with different values of $To1$ and $To2$ using $N = 95$ and $p = 10$. It is found that for GCD (without Bregman iteration), the sensitivity index error will not reduce

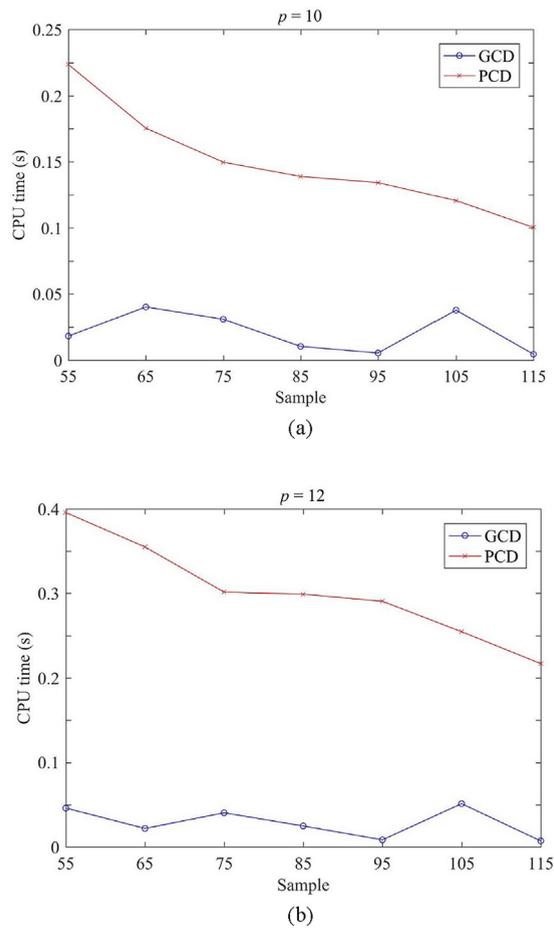


Fig. 4. Ishigami function–CPU time (s) for (a) $p = 10$ and (b) $p = 12$ with different samples.

any more as $Tol1$ decreases to 10^{-5} . However, for GCD with Bregman iteration (i.e. the proposed BGCD), the accuracy of estimated sensitivity indices improves for $Tol2 = 10^{-3}$ and 10^{-4} as compared with that for the GCD. Nevertheless, the needed Bregman iterations and resulting CPU time for $Tol2 = 10^{-4}$ are one to three orders of magnitude more than those for $Tol2 = 10^{-3}$ and 10^{-2} respectively. On the other hand, when $Tol2$ varies from 10^{-2} to 10^{-5} with $Tol1$ fixed respectively at 10^{-4} , 10^{-5} and 10^{-6} , the sensitivity index error has relatively lower values for $Tol2 = 10^{-4}$ to 10^{-2} . It is also noted that when $Tol1$ is bigger than $Tol2$ (e.g. $Tol1 = 10^{-3} > Tol2 = 10^{-4}$, $Tol1 = 10^{-4} > Tol2 = 10^{-5}$), the proposed BGCD produces large errors even with 1000 Bregman iterations. To have a better trade-off between the accuracy and efficiency for the BGCD, the following findings between $Tol1$ and $Tol2$ can be summarized from the above analysis: (1) the value of $Tol1$ should be less than that of $Tol2$, i.e. $Tol1 \leq Tol2$; (2) the ranges of $Tol1$ and $Tol2$ are recommended to be $10^{-6} \sim 10^{-4}$ and $10^{-3} \sim 10^{-2}$, respectively. These are in accordance with the partial functionality of Bregman iteration, which is devised to enhance the accuracy of GCD in just few iterations (e.g. 1 to 3 iterations for the Ishigami function). Therefore, the threshold value of $Tol2$ for Bregman iteration is logically set to be two to three orders of magnitude larger than that of $Tol1$ for GCD.

Third, effect of the regularization parameter λ on the accuracy, efficiency and sparsity of the sparse PCE by the BGCD is studied. Fig. 5 depicts the results of the sensitivity index error, CPU time and index of sparsity for λ varying within a broad range of $[10^{-3}, 10^5]$. It is observed that for λ smaller than 10^1 , although the CPU time and index of sparsity are desirable, the sensitivity index error increases rapidly as λ decreases. On the other hand, for λ bigger than 10^3 , although the sensitivity index error has a slower increase, the CPU time is one to two orders of magnitude more than that for λ smaller than 10^1 . The index of sparsity approaches to 0.95 if λ equals 2×10^4 , meaning that the level of sparsity is very minimum. To this end, an appropriate range of $[10^1, 10^3]$ for λ is recommended, which yields a good trade-off among the accuracy, efficiency and complexity (sparsity) for the developed sparse PCE. In this paper, for the purpose of simplicity, λ is proposed to be proportional to the dimensionality of the problem and degree of the PCE with a unified formulation: $\lambda = p \times n + 2$. It is worth noting that the accuracy of sensitivity indices can be further improved if the optimal value of λ based on the above parametric analysis is used, i.e. $\lambda = 100$ instead of $\lambda = 10 \times 3 + 2 = 32$ for the Ishigami function.

Table 2
Ishigami function—Comparison of sensitivity index error and CPU time with different threshold values using $N = 95$ and $p = 10$.

Algorithm	Tol1	Tol2	Sensitivity index error	Bregman iterations	CPU time (s)	
GCD	1.00E-03	-	1.43E-03	-	0.001	
	1.00E-04		9.51E-04		0.002	
	1.00E-05		9.12E-04		0.004	
	1.00E-06		9.26E-04		0.005	
	1.00E-07		9.28E-04		0.005	
BGCD	1.00E-03	1.00E-02	1.43E-03	1	0.002	
	1.00E-04	1.00E-02	9.51E-04	1	0.003	
	1.00E-05		9.12E-04	1	0.004	
	1.00E-06		9.26E-04	1	0.005	
	1.00E-07		9.28E-04	1	0.006	
	1.00E-03		1.00E-03	1.42E-03	3	0.004
	1.00E-04	1.00E-03	9.96E-04	2	0.004	
	1.00E-05		5.60E-04	2	0.009	
	1.00E-06		5.76E-04	2	0.012	
	1.00E-07		5.73E-04	2	0.015	
	1.00E-03		1.00E-04	1.32E-03	1000	0.949
	1.00E-04	1.00E-04	9.57E-04	44	0.125	
	1.00E-05		8.76E-04	31	0.248	
	1.00E-06		8.56E-04	26	0.616	
	1.00E-07		8.57E-04	26	1.149	
	1.00E-04		1.00E-02	9.51E-04	1	0.002
	1.00E-05	1.00E-05	1.00E-03	9.96E-04	2	0.004
			1.00E-04	9.57E-04	44	0.126
			1.00E-05	1.44E-03	1000	3.595
			1.00E-02	9.12E-04	1	0.004
			1.00E-03	5.60E-04	2	0.247
		1.00E-06	1.00E-04	8.76E-04	31	0.261
			1.00E-05	9.56E-04	157	1.849
			1.00E-02	9.26E-04	1	0.005
			1.00E-03	5.76E-04	2	0.012
1.00E-04			8.56E-04	26	0.616	
1.00E-07		1.00E-05	1.00E-03	1.00E-03	144	5.624

Lastly in this example, sensitivity analysis results in terms of e_s by the proposed BGCD are compared with two well-established greedy algorithms, i.e. OMP and LAR, for different degrees of the PCE ($p = 8, 9, \dots, 15$) using 95 samples, as presented in Table 3. It is obvious that the proposed BGCD has superior accuracy than both OMP and LAR for all eight cases while the performances of OMP and LAR vary with different degrees of the PCE. As for the index of sparsity, the BGCD always outperforms OMP and LAR for all degrees of the PCE. In terms of Bregman iterations, the maximum number is 8, and it ranges from 1 to 2 for higher degrees of the PCE such as $p = 10 \sim 15$. As a result, the CPU time by using BGCD is at similar level to that with LAR as p increases, which is a bit more than that by OMP. In addition, the Sobol' and total sensitivity indices obtained by sparse PCEs based on the proposed BGCD and two adaptive algorithms [45,10] are compared in Table 4. The number of model evaluations (N) and degree of the PCE (p) in the benchmark adaptive algorithm [45] are used for the BGCD, while the results by sparse PCE [10] with similar numbers of N and p are adopted for comparison. It is observed that the proposed BGCD outperforms two benchmark adaptive algorithms [45,10] for 7 and 10 out of 10 sensitivity indices, respectively. It is worth noting that, compared with the adaptive algorithms which sequentially selects the important basis polynomials from the full PCE, the proposed BGCD can efficiently build an optimal sparse PCE with the significant basis polynomials based on a one-time screening effort.

5.2. Example 2: Sobol' function

Let us consider now the Sobol' function [47]:

$$Y = \prod_{i=1}^n \frac{|4X_i - 2| + a_i}{1 + a_i}, \tag{34}$$

where the input variables $X_i, i = 1, 2, \dots, n$, are uniformly distributed over $[0, 1]$ and a_i 's are non-negative constants, gathered into the vector \mathbf{a} . For numerical application, the number of input variables is set as $n = 8$ with $\mathbf{a} = \{1, 2, 5, 10, 20, 50, 100, 500\}$, and the interaction order is set as $\eta = 4$. The sensitivity indices of Y can be derived analytically. Here, same as done in the previous example, they are estimated by postprocessing a sparse PCE of the model response Y . The sparse PC coefficients are evaluated using Sobol' quasi-random sequences with $N = 300, 350, \dots, 700$. The degree of the PCE is adopted as $p = 3, 4, \dots, 9$. As analysed in the previous example, the threshold vales for the BGCD are set as $Tol1 = 10^{-4}$ and $Tol2 = 10^{-2}$ in this example.

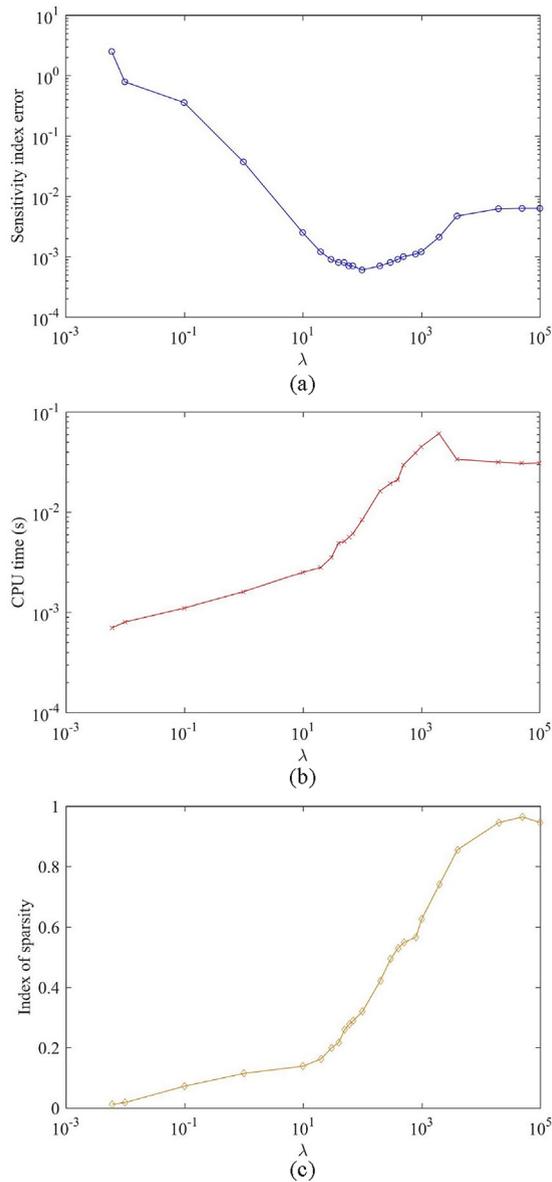


Fig. 5. Ishigami function—Effect of the regularization parameter λ on the (a) accuracy, (b) efficiency and (c) sparsity of the sensitivity indices by sparse PCE using the BGCD.

Convergence of the objective function by using GCD without Bregman iteration and PCD is illustrated in Fig. 6 for $p = 5$ and $N = 300$. It is found that by using GCD, the objective function reaches the final converged value of 8.376 in 490 iterations. However, PCD needs 48,324 iterations, which are around 99 times of those taken by GCD, to obtain the same result. Fig. 7 depicts comparison of the CPU time between GCD and PCD for $N = 300$ with different degrees of the PCE. It is observed that, as p varies from 3 to 7, the CPU time needed by PCD increases exponentially from 0.007 (7×10^{-3}) to 4401.372 ($\approx 4.401 \times 10^3$) s, while the time taken by GCD has a slight increment from 0.002 to 0.318 s only. Comparing the convergence and efficiency results between GCD and PCD for this and previous examples, it is believed that as the number of PC coefficients increases (e.g. from $\frac{(3+12)!}{3!12!} = 455$ for Ishigami function to $\frac{(8+5)!}{8!5!} = 1287$ for Sobol' function), the superiority of GCD over PCD will become more and more significant. This is one of the main motivations to develop GCD-based algorithm for building sparse PCE, and has also been validated with the following examples. Nevertheless, for the sake of brevity, comparisons of the convergence and efficiency between GCD and PCD will not be presented in the sequel.

Sensitivity index errors by the proposed BGCD are compared with those by OMP and LAR in Table 5 for different samples and degrees of the PCE. It is observed that BGCD outperforms both OMP and LAR for all cases of combinations with $N = 300$,

Table 3
Ishigami function–Sensitivity index errors with different degrees of the PCE using 95 samples.

Degree	Algorithm	Sensitivity index error	Index of sparsity	Bregman iterations	CPU time (s)
8	BGCD	0.0097	0.75	8	0.177
	OMP	0.0270	0.87	–	0.006
	LAR	0.0538	0.86	–	0.026
9	BGCD	0.0106	0.63	4	0.067
	OMP	0.0220	0.70	–	0.006
	LAR	0.0637	0.69	–	0.028
10	BGCD	0.0006	0.28	2	0.009
	OMP	0.0035	0.55	–	0.005
	LAR	0.0011	0.57	–	0.021
11	BGCD	0.0009	0.16	1	0.007
	OMP	0.0060	0.45	–	0.006
	LAR	0.0017	0.48	–	0.023
12	BGCD	0.0000	0.14	2	0.013
	OMP	0.0004	0.35	–	0.005
	LAR	0.0025	0.40	–	0.015
13	BGCD	0.0000	0.12	2	0.014
	OMP	0.0005	0.28	–	0.005
	LAR	0.0000	0.34	–	0.013
14	BGCD	0.0000	0.10	2	0.015
	OMP	0.0011	0.26	–	0.005
	LAR	0.0001	0.27	–	0.013
15	BGCD	0.0000	0.09	2	0.019
	OMP	0.0005	0.23	–	0.008
	LAR	0.0012	0.26	–	0.018

Table 4
Ishigami function–Estimates of the sensitivity indices by sparse PCEs based on the proposed BGCD and two adaptive algorithms.

Sensitivity indices	Analytical results	The proposed BGCD	Sparse PCE [45]	Sparse PCE [10]
S_1	0.31	0.31	0.32	0.29
S_2	0.44	0.45	0.44	0.43
S_3	0.00	0.00	0.00	0.00
$S_{1,2}$	0.00	0.00	0.00	0.01
$S_{1,3}$	0.24	0.24	0.25	0.25
$S_{2,3}$	0.00	0.00	0.00	0.00
$S_{1,2,3}$	0.00	0.00	0.00	0.01
S_1^T	0.56	0.55	0.56	0.57
S_2^T	0.44	0.45	0.44	0.46
S_3^T	0.24	0.24	0.23	0.27
Model evaluations		58	58	84
Degree of the PCE		9	9	7

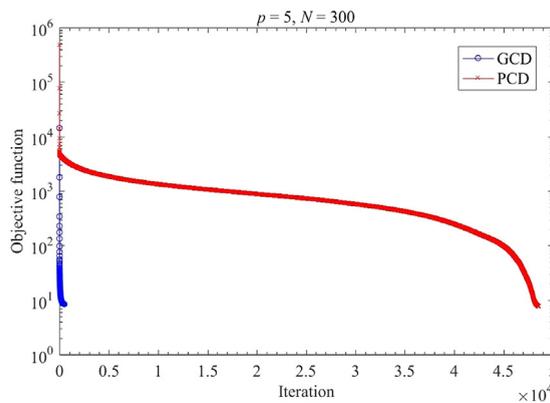


Fig. 6. Sobol' function–Convergence of the objective function for $p = 5$ and $N = 300$.

350, ..., 700 and $p = 5, 7, 9$ except for the case of $N = 500$ and $p = 7$. Also, LAR has better accuracy than OMP for most cases as in the previous example. The values of index of sparsity IS by using BGCD are at similar levels to those by using OMP and LAR

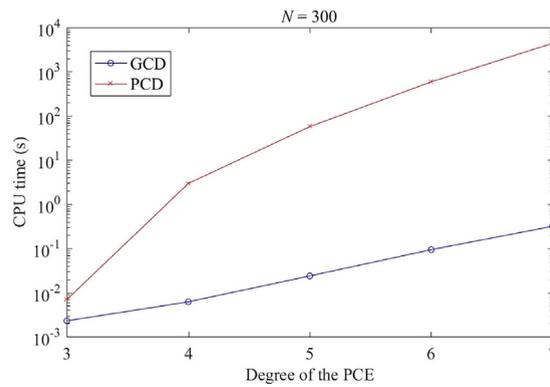


Fig. 7. Sobol' function–Comparison of CPU time between GCD and PCD for $N = 300$ with different degrees of the PCE.

in this example. Furthermore, the number of Bregman iterations used in the BGCD is 2 for the case of $N = 700$ and $p = 5$, and 1 for other 26 cases. Comparisons of CPU time among BGCD, OMP and LAR for $p = 5, 7$ and 9 with different samples are shown in Fig. 8. It is found that as the number of samples increases, the CPU time taken by BGCD has very minimal change. On the contrary, there is a considerable increase of the CPU time spent by both OMP and LAR, especially LAR. As a result, the superiority of BGCD over OMP and LAR in computational efficiency becomes more notable as the number of samples increases. Moreover, the results of Sobol' and total sensitivity indices obtained by sparse PCEs based on the proposed BGCD and two adaptive algorithms [45,10] are listed in Table 6. It is observed that with same or similar numbers of N and p , the proposed BGCD outperforms two benchmark adaptive algorithms for 15 out of 16 sensitivity indices. It is worthy to note that the superiority of the BGCD over the adaptive algorithms is also significant for the sensitivity indices (e.g. $S_3^T-S_8^T$) with relatively smaller values.

5.3. Example 3: Morris function

In order to assess the proposed algorithm for high-dimensional problems, we consider now the so-called Morris function [16]:

$$Y = b_0 + \sum_{i=1}^{20} b_i X_i + \sum_{i<j}^{20} b_{ij} X_i X_j + \sum_{i<j<k}^{20} b_{ijk} X_i X_j X_k + \sum_{i<j<k<l}^{20} b_{ijkl} X_i X_j X_k X_l, \tag{35}$$

where

Table 5
Sobol' function–Sensitivity analysis results with different samples and degrees of the PCE.

	N	Degree of the PCE = 5			Degree of the PCE = 7			Degree of the PCE = 9		
		BGCD	OMP	LAR	BGCD	OMP	LAR	BGCD	OMP	LAR
e_s	300	0.0520	0.4411	0.0865	0.0630	0.0944	0.1601	0.0269	0.0766	0.0623
	350	0.0504	0.3634	0.0919	0.0454	0.1895	0.0725	0.0397	0.0500	0.0567
	400	0.0636	0.4102	0.1729	0.0218	0.1236	0.0588	0.0215	0.0638	0.0307
	450	0.0735	0.4908	0.1292	0.0358	0.1188	0.0716	0.0236	0.0614	0.1466
	500	0.0775	0.5253	0.1574	0.0502	0.1475	0.0450	0.0170	0.0883	0.0235
	550	0.0851	0.6885	0.8483	0.0462	0.2035	0.1257	0.0290	0.0841	0.3972
	600	0.1065	0.5612	0.2614	0.0410	0.1576	0.0554	0.0210	0.0569	0.0231
IS	650	0.1039	0.6595	0.3185	0.0273	0.2139	1.8796	0.0219	0.0719	0.0697
	700	0.1913	0.8489	0.5561	0.0217	0.2056	0.0579	0.0135	0.0668	0.0163
	300	0.29	0.24	0.24	0.07	0.06	0.06	0.02	0.02	0.02
	350	0.33	0.28	0.28	0.08	0.07	0.07	0.02	0.02	0.02
	400	0.36	0.32	0.32	0.09	0.08	0.08	0.03	0.03	0.03
	450	0.41	0.36	0.37	0.09	0.09	0.09	0.03	0.03	0.03
	500	0.41	0.40	0.41	0.10	0.10	0.10	0.03	0.03	0.03
	550	0.47	0.44	0.45	0.12	0.11	0.11	0.03	0.04	0.04
	600	0.52	0.48	0.49	0.14	0.12	0.12	0.04	0.04	0.04
	650	0.54	0.52	0.53	0.16	0.13	0.13	0.05	0.04	0.04
700	0.64	0.56	0.57	0.15	0.13	0.14	0.05	0.05	0.05	

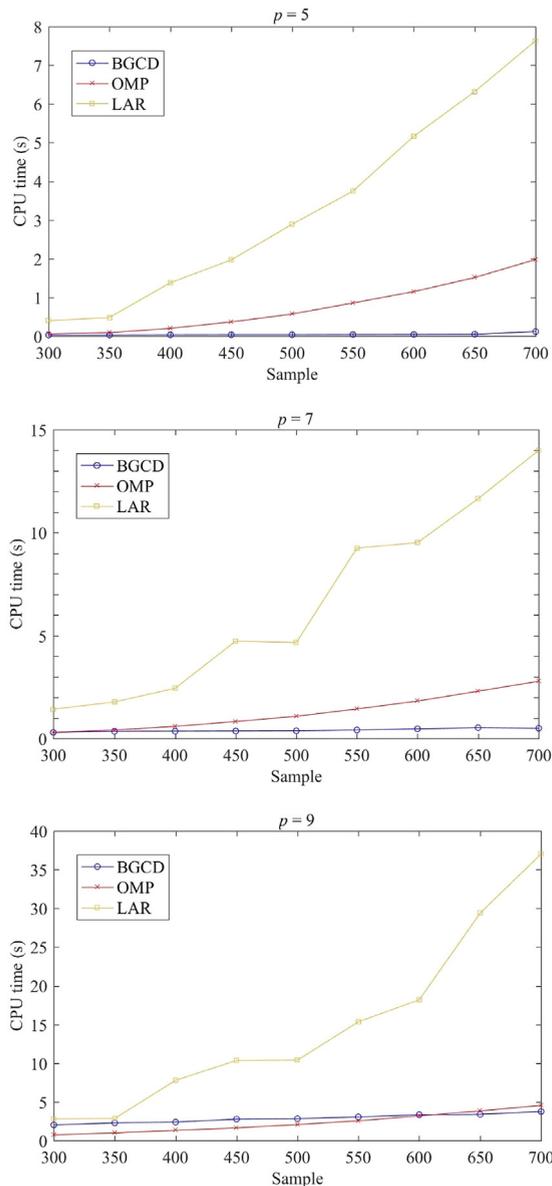


Fig. 8. Sobol' function—Comparison of CPU time among BGCD, OMP and LAR for $p = 5, 7$ and 9 with different samples.

$$X_i = \begin{cases} 2(1.1x_i/(x_i + 0.1) - 0.5) & \text{if } i = 3, 5, 7 \\ 2(x_i - 0.5) & \text{otherwise} \end{cases} \quad (36)$$

and x_i ($i = 1, 2, \dots, 20$) are uniformly distributed over $[0, 1]$. The coefficients b_i are assigned as follows:

$$\begin{cases} b_i = 20 & \text{for } i = 1, \dots, 10 \\ b_{ij} = -15 & \text{for } i, j = 1, \dots, 6 \\ b_{ijk} = -10 & \text{for } i, j, k = 1, \dots, 5 \\ b_{ijkl} = 5 & \text{for } i, j, k, l = 1, \dots, 4. \end{cases} \quad (37)$$

The remaining coefficients are defined by $b_0 = 0, b_i = (-1)^i, b_{ij} = (-1)^{i+j}$ and $b_{ijk} = b_{ijkl} = 0$. Unlike the previous examples, the Morris function is neither even or odd, and its dimensionality is high. Therefore, to make the computational cost affordable, the total degree of the PCE and interaction order is set as $p = 3$ and $\eta = 3$. The sensitivity indices are computed by post-processing the sparse PCE, which is built using Sobol' quasi-random sequences with $N = 200, 300, 400, 500$ and 900 . The

Table 6
Sobol' function—Estimates of the sensitivity indices by sparse PCEs based on the proposed BGCD and two adaptive algorithms.

Sensitivity indices	Analytical results	The proposed BGCD	Sparse PCE [45]	Sparse PCE [10]
S_1	0.60	0.59	0.59	0.56
S_2	0.27	0.27	0.25	0.22
S_3	0.07	0.07	0.07	0.05
S_4	0.02	0.02	0.01	0.02
S_5	0.01	0.01	0.01	0.01
S_6	0.00	0.00	0.00	0.00
S_7	0.00	0.00	0.00	0.00
S_8	0.00	0.00	0.00	0.00
S_1^T	0.63	0.61	0.63	0.59
S_2^T	0.29	0.29	0.28	0.26
S_3^T	0.08	0.08	0.10	0.10
S_4^T	0.02	0.03	0.03	0.05
S_5^T	0.01	0.01	0.03	0.03
S_6^T	0.00	0.01	0.02	0.04
S_7^T	0.00	0.01	0.02	0.03
S_8^T	0.00	0.01	0.01	0.03
Model evaluations		127	127	150
Degree of the PCE		4	4	6

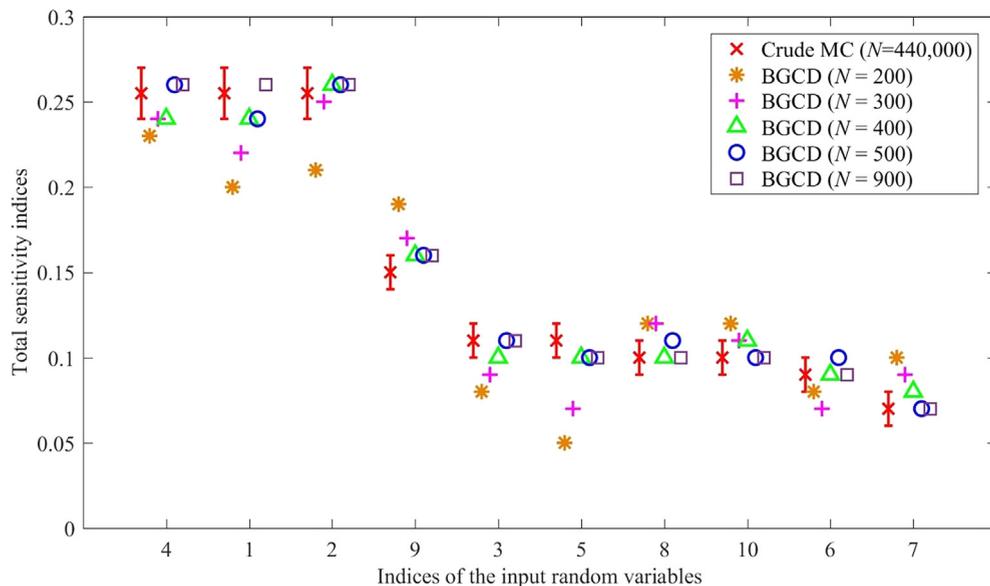


Fig. 9. Morris function—Estimates of the ten greatest total sensitivity indices with different numbers of model evaluations by using the proposed BGCD.

threshold values for the proposed BGCD are defined as $Tol1 = 10^{-4}$ and $Tol2 = 10^{-3}$. Herein, due to the high dimensionality of Morris function, this example adopts a smaller threshold value of $Tol2 = 10^{-3}$ as compared with $Tol2 = 10^{-2}$ for the Sobol' function.

Because there are no reference values for the Morris function, the quality of the estimation is evaluated by comparing the sensitivity indices and their credible intervals. The 95% confidence intervals of these sensitivity indices are computed based on crude MC simulation with 440,000 samples [10]. Estimates of the ten greatest total sensitivity indices with different numbers of model evaluations by using the proposed BGCD are plotted in Fig. 9. It is observed that as the number N of model evaluations increases, almost all ten total sensitivity indices come close to the 95% confidence intervals. When N is equal to 400, all ten total sensitivity indices are within the 95% confidence intervals, which allows one to correctly distinguish three groups of input variables, namely, x_1, x_2 and x_4 are important variables, x_9 is intermediately important, and x_3, x_5-x_8 , and $x_{10}-x_{20}$ are of little significance. As N further increases to 900, 8 out of 10 total sensitivity indices are around the centers of the 95% confidence intervals.

Results of the ten greatest total sensitivity indices by using BGCD, OMP and LAR are compared in Table 7. It is shown that the proposed BGCD yields the most accurate results of ten greatest total sensitivity indices which are all included in the MC 95% confidence intervals with both $N = 400$ and $N = 500$. Unlike the previous examples, LAR is less accurate than OMP in this

Table 7
Morris function–Results of the ten greatest total sensitivity indices.

Sensitivity indices	BGCD		OMP		LAR		MC 95% confidence intervals
S_4^T	0.24	0.26	0.24	0.25	0.24	0.24	[0.24, 0.27]
S_1^T	0.24	0.24	0.23*	0.22*	0.22*	0.23*	[0.24, 0.27]
S_2^T	0.26	0.26	0.23*	0.24	0.23*	0.24	[0.24, 0.27]
S_9^T	0.16	0.16	0.16	0.16	0.16	0.17*	[0.14, 0.16]
S_3^T	0.10	0.11	0.11	0.10	0.09*	0.09*	[0.10, 0.12]
S_5^T	0.10	0.10	0.10	0.10	0.08*	0.08*	[0.10, 0.12]
S_8^T	0.10	0.11	0.11	0.10	0.12*	0.11	[0.09, 0.11]
S_{10}^T	0.11	0.10	0.10	0.11	0.13*	0.13*	[0.09, 0.11]
S_6^T	0.09	0.10	0.09	0.10	0.08	0.09	[0.08, 0.10]
S_7^T	0.08	0.07	0.07	0.07	0.08	0.07	[0.06, 0.08]
Model evaluations	400	500	400	500	400	500	440,000
Bregman iterations	1	1	–	–	–	–	–
CPU time (s)	0.357	0.319	0.263	0.689	0.990	2.258	–
Index of sparsity	0.56	0.70	0.22	0.27	0.23	0.28	–

* Results excluded from the MC 95% confidence intervals.

Table 8
Morris function–Estimates of the ten greatest total sensitivity indices by sparse PCEs based on the proposed BGCD and two adaptive algorithms.

Sensitivity indices	MC 95% confidence intervals	The proposed BGCD	Sparse PCE [45]	Sparse PCE [10]
S_4^T	[0.24, 0.27]	0.26	0.25	0.25
S_1^T	[0.24, 0.27]	0.24	0.25	0.25
S_2^T	[0.24, 0.27]	0.25	0.24	0.24
S_9^T	[0.14, 0.16]	0.16	0.16	0.15
S_3^T	[0.10, 0.12]	0.10	0.10	0.11
S_5^T	[0.10, 0.12]	0.10	0.11	0.10
S_8^T	[0.09, 0.11]	0.11	0.11	0.10
S_{10}^T	[0.09, 0.11]	0.10	0.11	0.11
S_6^T	[0.08, 0.10]	0.10	0.08	0.09
S_7^T	[0.06, 0.08]	0.07	0.08	0.07
Model evaluations		520	520	750
Degree of the PCE		3	4	11

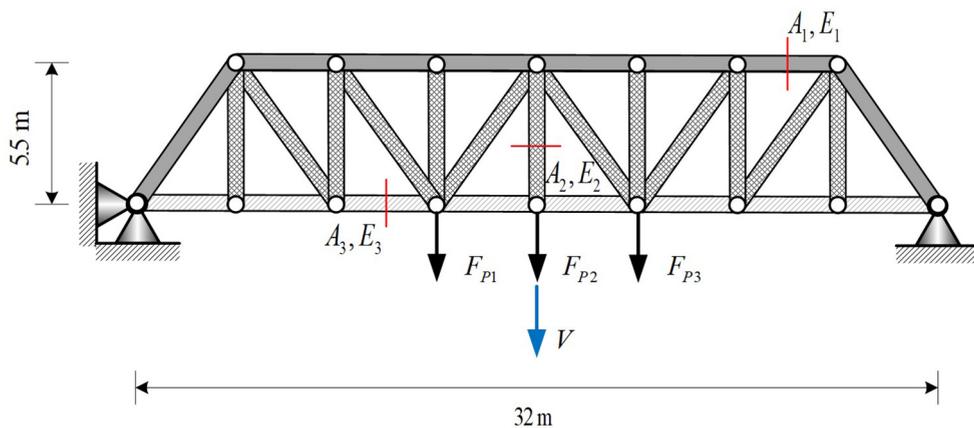


Fig. 10. Schematic diagram of a planar truss structure.

example. The number of Bregman iterations for the BGCD is 1 with both cases of model evaluations. In terms of computational efficiency, as the number of model evaluations increases from 400 to 500, the CPU time required by the proposed BGCD decreases slightly from 0.357 to 0.319 s, whereas that taken by OMP and LAR increases from 0.263 and 0.990 s to 0.689 and 2.258 s, respectively. This is in accordance with the findings in the previous examples. The level of sparsity is also depicted in Table 7. Although OMP and LAR produce sparser structures in this example, the proposed algorithm still yields a

Table 9
A planar truss structure–Input random variables with distribution parameters.

Input variable	Distribution	Mean	Coefficient of variation
A_1	Normal	$2.5 \times 10^{-3} m^2$	0.1
A_2	Normal	$1.185 \times 10^{-3} m^2$	0.1
A_3	Normal	$3.031 \times 10^{-3} m^2$	0.1
E_1, E_2, E_3	Normal	$2.1 \times 10^{11} Pa$	0.1
F_{P1}, F_{P2}, F_{P3}	Normal	$5.0 \times 10^4 N$	0.15

Table 10
A planar truss structure–Results of the Sobol' and total sensitivity indices.

Variables	Sobol' indices				Total sensitivity indices			
	MC	BGCD	OMP	LAR	MC	BGCD	OMP	LAR
A_1	0.086	0.085	0.086*	0.088	0.087	0.086*	0.088*	0.090
E_1	0.086	0.086*	0.084	0.084	0.088	0.087*	0.086	0.087*
A_2	0.132	0.129	0.130*	0.136	0.135	0.131	0.132*	0.140
E_2	0.132	0.132*	0.136	0.134	0.135	0.135*	0.138	0.138
A_3	0.030	0.030*	0.030*	0.029	0.031	0.030	0.031*	0.031*
E_3	0.030	0.030*	0.028	0.031	0.031	0.030*	0.029	0.033
F_{P1}	0.130	0.130*	0.129	0.124	0.130	0.130*	0.130*	0.126
F_{P2}	0.238	0.241*	0.242	0.234	0.241	0.243*	0.243*	0.238
F_{P3}	0.129	0.131	0.128	0.129*	0.130	0.132	0.129	0.130*
Model evaluations	4.4×10^6	130	130	130	4.4×10^6	130	130	130
Bregman iterations	–	3	–	–	–	3	–	–
CPU time (s)	3.8×10^3	0.075	0.015	0.025	3.8×10^3	0.075	0.015	0.025
Index of sparsity	–	0.28	0.55	0.59	–	0.28	0.55	0.59

* Best results among the compared algorithms: BGCD, OMP and LAR.

more accurate PCE. This demonstrates the proposed BGCD provides a better trade-off between model accuracy and complexity. Also, Table 8 presents the estimates of the ten greatest total sensitivity indices obtained by sparse PCEs based on the proposed BGCD and two adaptive algorithms [45,10]. The sensitivity indices obtained by the BGCD are closer to the centers of the MC 95% confidence intervals for 8 out of 10 cases than those by Zhou et al. [45] with same/similar numbers of N and p , and comparable to those by Blatman and Sudret [10] in which much larger numbers of N (i.e. $750 \gg 520$) and p (i.e. $11 \gg 3$) were adopted.

5.4. Example 4: A planar truss structure

Let us consider a planar truss structure [58] sketched in Fig. 10, which comprises 14 horizontal, 7 vertical and 8 diagonal bars. Three vertical loads are applied on the lower central portion of the truss. A finite element model made of 29 bar elements is used. Nine parameters are assumed to be random and modelled by independent input random variables, namely the cross-sectional areas and Young's moduli of the upper horizontal and outer diagonal bars, the vertical and inner diagonal bars, and the lower horizontal bars (respectively denoted by A_1, E_1, A_2, E_2 and A_3, E_3) and the applied loads (denoted by F_{P1}, F_{P2}, F_{P3}), whose mean and coefficient of variation are given in Table 9. The sensitivity of the maximum deflection V of the truss structure to each input variable is investigated. To this end, the Sobol' indices and total sensitivity indices are derived by postprocessing the sparse PCE built using Sobol' quasi-random sequences with $N = 130, p = 3$ and $\eta = 3$. Same as those used for the Sobol' function, the threshold values for the proposed BGCD are set as Tol 1 = 10^{-6} and Tol 2 = 10^{-3} in this example.

Results of the Sobol' and total sensitivity indices by using BGCD, OMP and LAR are compared in Table 10. Reference results are obtained using crude MC simulation (4.4×10^6 finite element runs are performed as a whole). It is observed from Table 10 that the sensitivity indices associated with A_1 and E_1 (respectively A_2 and E_2, A_3 and E_3) are similar, due to the fact that these variables have the same type of probability density function and coefficient of variation and the deflection V only depends on them through the products E_1A_1, E_2A_2 and E_3A_3 . The symmetry of the problem can also be reflected from the sensitivity indices by giving similar significances to the symmetrically applied loads (e.g. F_{P1} and F_{P3}), and greater sensitivity index is logically attributed to the load (e.g. F_{P2}) imposed on the midspan point than the loads away from it. The differences between the total sensitivity indices and Sobol' indices indicate small interactive effects of various input variables (i.e. low interaction order). It can be concluded that the proposed BGCD yields the most accurate results for 12 out of 18 sensitivity indices, while OMP is the second best. The level of sparsity generated by using BGCD outperforms that by using OMP and LAR in this example.

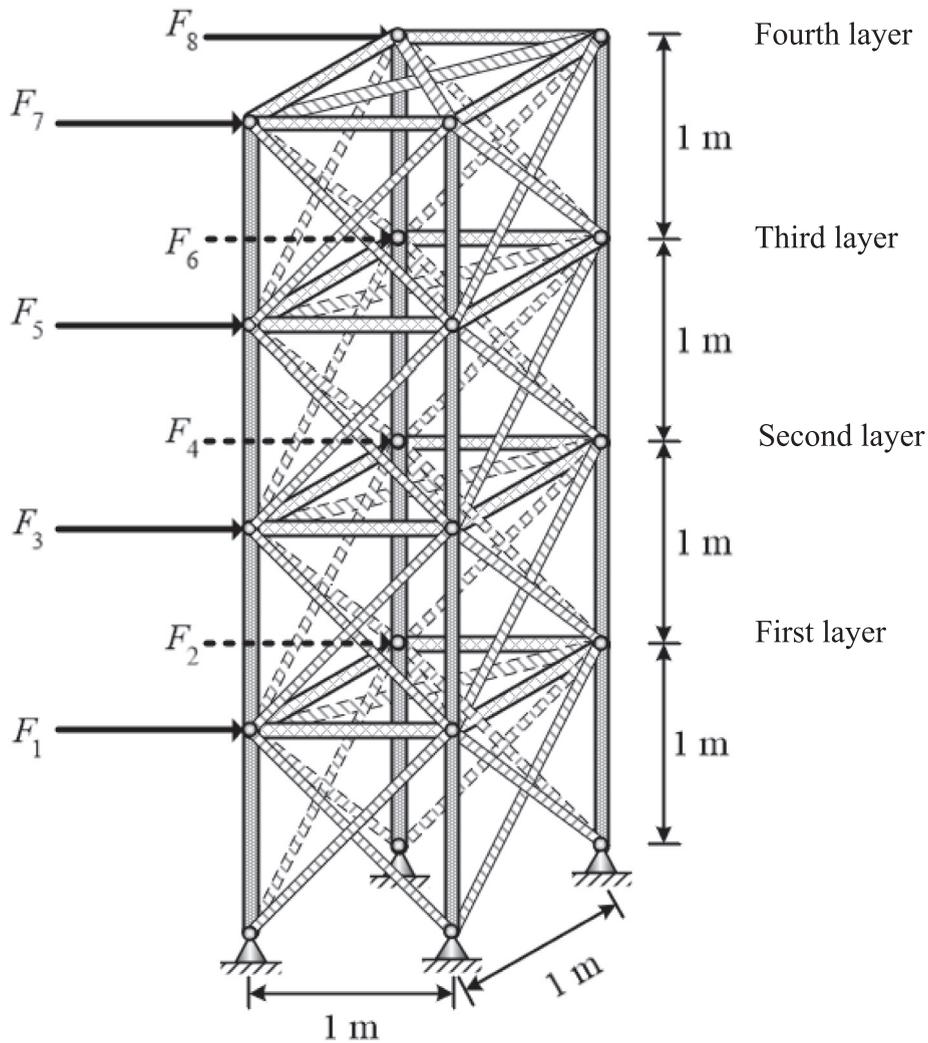


Fig. 11. Schematic diagram of a spatial truss structure.

The sparse PCE constructed by the proposed BGCD with 130 model evaluations provides comparable sensitivity results with those by crude MC simulation ($N = 4.4 \times 10^6$). The number of Bregman iterations used is 3 for $N = 130$. In terms of computational time, it appears that more than one hour (i.e. 3.8×10^3 s) is needed for crude MC simulation with 4.4×10^6 model evaluations. Therefore, the CPU time of building a sparse PCE is negligible compared to the cost of finite element simulations. This demonstrates the practicability of the proposed algorithm for engineering applications.

5.5. Example 5: a spatial truss structure

The last example considers a 72-bar spatial truss structure [59] under lateral loads, as shown in Fig. 11. A finite element model made of 72 bar elements is used. Twenty-one parameters are modelled by independent random variables: the length of single horizontal or vertical bar (L), the Young's moduli and cross-sectional areas of the vertical, horizontal, and first layer to fourth layer diagonal bars (denoted by E_1, A_1 to E_6, A_6 respectively), and the applied wind loads (F_1 - F_8), whose mean and coefficient of variation are given in Table 11. The sensitivity of the maximum horizontal displacement of the truss structure to each input variable is investigated. The Sobol' and total sensitivity indices are derived by postprocessing the sparse PCE built using Sobol' quasi-random sequences with $N = 600, p = 3$ and $\eta = 3$. The threshold values for the proposed BGCD are set as $Tol1 = 10^{-6}$ and $Tol2 = 10^{-3}$.

Results of the Sobol' and total sensitivity indices by using BGCD, OMP and LAR are presented in Table 12. Reference results are obtained using crude MC simulation with 18.4×10^7 finite element runs. Same as the phenomenon in Example 4, the

Table 11
A spatial truss structure–Input random variables with distribution parameters.

Input variable	Distribution	Mean	Coefficient of variation
L	Normal	1 m	0.1
E_1	Lognormal	2.1×10^{11} Pa	0.1
A_1	Lognormal	3.0×10^{-3} m ²	0.1
E_2	Lognormal	2.1×10^{11} Pa	0.1
A_2	Lognormal	3.0×10^{-3} m ²	0.1
E_3	Lognormal	2.1×10^{11} Pa	0.1
A_3	Lognormal	3.0×10^{-3} m ²	0.1
E_4	Lognormal	2.1×10^{11} Pa	0.1
A_4	Lognormal	3.0×10^{-3} m ²	0.1
E_5	Lognormal	2.1×10^{11} Pa	0.1
A_5	Lognormal	3.0×10^{-3} m ²	0.1
E_6	Lognormal	2.1×10^{11} Pa	0.1
A_6	Lognormal	3.0×10^{-3} m ²	0.1
F_1	Weibull	3.0×10^5 N	0.15
F_2	Weibull	3.0×10^5 N	0.15
F_3	Weibull	4.0×10^5 N	0.15
F_4	Weibull	4.0×10^5 N	0.15
F_5	Weibull	5.0×10^5 N	0.15
F_6	Weibull	5.0×10^5 N	0.15
F_7	Weibull	6.0×10^5 N	0.15
F_8	Weibull	6.0×10^5 N	0.15

Table 12
A spatial truss structure–Results of the Sobol' and total sensitivity indices.

Variables	Sobol' indices				Total sensitivity indices			
	MC	BGCD	OMP	LAR	MC	BGCD	OMP	LAR
L	0.418	0.417*	0.416	0.409	0.421	0.423	0.422*	0.417
E_1	0.183	0.181*	0.181*	0.176	0.185	0.184*	0.184*	0.181
A_1	0.182	0.182*	0.182*	0.178	0.185	0.185*	0.186	0.183
E_2	0.001	0.001*	0.001*	0.001*	0.001	0.001*	0.001*	0.003
A_2	0.001	0.001*	0.001*	0.001*	0.001	0.001*	0.001*	0.003
E_3	0.009	0.010*	0.010*	0.010*	0.010	0.010*	0.010*	0.012
A_3	0.010	0.009	0.010*	0.009	0.010	0.010*	0.011	0.011
E_4	0.002	0.002*	0.002*	0.003	0.002	0.002*	0.002*	0.005
A_4	0.002	0.002*	0.002*	0.002*	0.002	0.002*	0.002*	0.004
E_5	0.001	0.001*	0.001*	0.001*	0.001	0.001*	0.001*	0.003
A_5	0.001	0.001*	0.001*	0.001*	0.001	0.001*	0.001*	0.003
E_6	0.000	0.000*	0.000*	0.000*	0.000	0.000*	0.000*	0.001
A_6	0.000	0.000*	0.000*	0.000*	0.000	0.000*	0.000*	0.002
F_1	0.000	0.000*	0.000*	0.000*	0.000	0.000*	0.000*	0.002
F_2	0.000	0.000*	0.000*	0.000*	0.000	0.000*	0.000*	0.001
F_3	0.003	0.003*	0.003*	0.004	0.004	0.003	0.004*	0.006
F_4	0.003	0.004	0.004	0.003*	0.004	0.004*	0.004*	0.006
F_5	0.020	0.020*	0.019	0.019	0.020	0.020*	0.020*	0.022
F_6	0.020	0.020*	0.019	0.018	0.021	0.020*	0.020*	0.022
F_7	0.069	0.069*	0.069*	0.072	0.071	0.071*	0.070	0.077
F_8	0.069	0.069*	0.068	0.068	0.071	0.070	0.070	0.071*
Model evaluations	1.84×10^7	600	600	600	1.84×10^7	600	600	600
Bregman iterations	–	3	–	–	–	3	–	–
CPU time (s)	3.4×10^4	0.990	1.235	12.176	3.4×10^4	0.990	1.235	12.176
Index of sparsity	–	0.158	0.272	0.090	–	0.158	0.272	0.090

* Best results among the compared algorithms: BGCD, OMP and LAR.

sensitivity indices associated with E_1 and A_1 (respectively E_2 and A_2 to E_6 and A_6) are similar. Sensitivity indices associated with F_1 to F_8 reflect the symmetry of the problem by giving similar significances to the symmetrical load pairs (e.g. F_1 and F_2 , F_3 and F_4 , F_5 and F_6 , F_7 and F_8), and gradually increase as the load pairs moves from the first to fourth layer of the truss structure. Greater sensitivity indices are logically attributed to the Young's modulus and cross-sectional area of the vertical bar than those of the horizontal and diagonal bars. The differences between Sobol' and total sensitivity indices are trivial, which indicates small interactive effects among the input variables. Taking the MC simulation results as analytical values, the sen-

sitivity index error e_s by using BGCD, OMP and LAR is obtained as 0.011, 0.017 and 0.076, respectively. Therefore, it is concluded that the proposed BGCD yields the most accurate results while OMP outperforms LAR. The research findings of this spatial truss structure with high-dimensional (twenty-one) input variables are in good accordance with the planar truss structure in Example 4, which demonstrates further the great potential of the proposed algorithm in practical engineering.

6. Conclusions

In this paper, a novel algorithm based on the integration of GCD and Bregman iteration is proposed for building sparse PCE of model response. It is then used to compute Sobol' indices for GSA. Taking advantage of the efficiency of GCD in sparsity exploitation and capability of Bregman iteration in accuracy enhancement, the proposed BGCD efficiently and accurately selects the significant basis polynomials by solving a LASSO-based regression problem (also known as general ℓ_1 -minimization problem) of the PC coefficients. The novelty and strength of the proposed algorithm are three-fold: (1) building sparse PCE by reducing model complexity with the LASSO-based regression and taking advantage of the efficiency and robustness of GCD in sparsity exploitation, (2) settling the inefficiency in selecting the regularization parameter in the LASSO-based regression and enhancing the accuracy of GCD simultaneously by incorporating Bregman iteration to form a hierarchical algorithm structure, and (3) handling problems with far more unknown coefficients than the number of model evaluations.

The performance of the proposed algorithm is demonstrated by five numerical examples. Several sparse reconstruction techniques including PCD, GCD, OMP, LAR and two adaptive algorithms are used to make comparisons with the proposed BGCD. It is shown that the proposed algorithm yields more accurate sensitivity results than those obtained by OMP, LAR and two adaptive algorithms, while the performances of the benchmark algorithms are not consistent. The superiority of BGCD over OMP and LAR in computational efficiency becomes notable as the number of samples increases. In terms of the level of sparsity, it is found that the proposed algorithm produces sparser PCE than that by OMP and LAR for problems with low interaction order such as the Ishigami function and maximum deflection of a truss structure. For problems with relatively high interaction order (e.g. the Morris function), although OMP and LAR yield sparser PCEs, the proposed algorithm still outperforms in terms of accuracy, demonstrating a better technique to maintain a balance between model accuracy and complexity.

The proposed algorithm aims at screening important basis polynomials and building an optimal sparse PCE with a fixed degree of the PCE. Moreover, although the main focus of this paper is to develop a novel BGCD algorithm for sparse PCE metamodeling with application to GSA, the constructed PCE model can be used for other applications such as estimation of the response probability density function, reliability analysis and so on. Some preliminary results for estimation of the response probability density function and reliability analysis have been obtained by the authors, which are not presented herein since they are out of the scope of this paper. Further work on the adaptive strategies for choosing the PCE degree and other usages of the developed sparse PCE is in progress.

CRedit authorship contribution statement

Jian Zhang: Supervision, Conceptualization, Methodology, Writing - original draft, Writing - review & editing. **Xinxin Yue:** Methodology, Investigation, Software, Data curation, Writing - review & editing. **Jiajia Qiu:** Investigation, Writing - review & editing. **Lijun Zhuo:** Writing - review & editing. **Jianguo Zhu:** Writing - review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (Grant Nos. 11872190, 11972014), the Six Talent Peaks Project in Jiangsu Province (Grant No. 2017-KTHY-010), and the Research Start-up Foundation for Jinshan Distinguished Professorship at Jiangsu University (Grant No. 4111480003).

References

- [1] W. Chen, R.C. Jin, A. Sudjianto, Analytical variance-based global sensitivity analysis in simulation-based design under uncertainty, *J. Mech. Des.* 127 (5) (2005) 875–886.
- [2] Z.P. Wu, D.H. Wang, P. Okolo N, F. Hu, W.H. Zhang, Global sensitivity analysis using a Gaussian radial basis function metamodel, *Reliab. Eng. Syst. Safety* 154 (2016) 171–179.
- [3] Z.X. Yuan, P. Liang, T. Silva, K.P. Yu, J.E. Mottershead, Parameter selection for model updating with global sensitivity analysis, *Mech. Syst. Sig. Process.* 115 (2019) 483–496.
- [4] K. Cheng, Z.Z. Lu, Y. Zhen, Multi-level multi-fidelity sparse polynomial chaos expansion based on Gaussian process regression, *Comput. Methods Appl. Mech. Eng.* 349 (2019) 360–377.

- [5] S. Li, B. Yang, F. Qi, Accelerate global sensitivity analysis using artificial neural network algorithm: case studies for combustion kinetic model, *Combust. Flame* 168 (2016) 53–64.
- [6] K. Cheng, Z.Z. Lu, Y.H. Wei, Y. Shi, Y.C. Zhou, Mixed kernel function support vector regression for global sensitivity analysis, *Mech. Syst. Sig. Process.* 96 (2017) 201–214.
- [7] J.W. Feng, L. Liu, D. Wu, G.Y. Li, M. Beer, W. Gao, Dynamic reliability analysis using the extended support vector regression (X-SVR), *Mech. Syst. Sig. Process.* 126 (2019) 368–391.
- [8] J. Zhang, X.X. Yue, J.J. Qiu, M.Y. Zhang, X.M. Wang, A unified ensemble of surrogates with global and local measures for global metamodeling, *Eng. Optim.* 53 (3) (2021) 474–495.
- [9] B. Sudret, Global sensitivity analysis using polynomial chaos expansions, *Reliab. Eng. Syst. Saf.* 93 (2008) 964–979.
- [10] G. Blatman, B. Sudret, Efficient computation of global sensitivity indices using sparse polynomial chaos expansions, *Reliab. Eng. Syst. Saf.* 95 (2010) 1216–1229.
- [11] E. Jacquelin, M.I. Friswell, S. Adhikari, O. Dessombz, J.J. Sinou, Polynomial chaos expansion with random and fuzzy variables, *Mech. Syst. Sig. Process.* 75 (2016) 41–56.
- [12] H.Z. Dai, Z.B. Zheng, H.H. Ma, An explicit method for simulating non-Gaussian and non-stationary stochastic processes by Karhunen-Loeve and polynomial chaos expansion, *Mech. Syst. Sig. Process.* 115 (2019) 1–13.
- [13] P.H. Ni, Y. Xia, J. Li, H. Hao, Using polynomial chaos expansion for uncertainty and sensitivity analysis of bridge structures, *Mech. Syst. Sig. Process.* 119 (2019) 293–311.
- [14] W. Zhao, L.Z. Bu, Global sensitivity analysis with a hierarchical sparse metamodeling method, *Mech. Syst. Sig. Process.* 115 (2019) 769–781.
- [15] H.P. Wan, W.X. Ren, M.D. Todd, Arbitrary polynomial chaos expansion method for uncertainty quantification and global sensitivity analysis in structural dynamics, *Mech. Syst. Sig. Process.* 142 (2020) 106732.
- [16] A. Saltelli, K. Chan, E.M. Scott, *Sensitivity Analysis*, John Wiley and Sons, Chichester, 2000.
- [17] R.G. Ghanem, P.D. Spanos, *Stochastic Finite Elements: A Spectral Approach*, Springer-Verlag, Berlin, 1991.
- [18] N. Wiener, The homogeneous chaos, *Am. J. Mathem.* 60 (4) (1938) 897–936.
- [19] D. Xiu, G.E. Karniadakis, The Wiener-Askey polynomial chaos for stochastic differential equations, *SIAM J. Sci. Comp.* 24 (2) (2002) 619–644.
- [20] H.G. Matthies, A. Keese, Galerkin methods for linear and nonlinear elliptic stochastic partial differential equations, *Comput. Methods Appl. Mech. Eng.* 194 (12–16) (2005) 1295–1331.
- [21] O.P. Le Maître, M.T. Reagan, H.N. Najm, R.G. Ghanem, O.M. Knio, A stochastic projection method for fluid flow: II. Random process, *J. Comput. Phys.* 181 (1) (2002) 9–44.
- [22] M. Berveiller, B. Sudret, M. Lemaire, Stochastic finite element: a non intrusive approach by regression, *Eur. J. Comput. Mech.* 15 (1–3) (2006) 81–92.
- [23] G. Blatman, B. Sudret, An adaptive algorithm to build up sparse polynomial chaos expansions for stochastic finite element analysis, *Probab. Eng. Mech.* 25 (2010) 183–197.
- [24] G. Blatman, B. Sudret, Adaptive sparse polynomial chaos expansion based on least angle regression, *J. Comput. Phys.* 230 (2011) 2345–2367.
- [25] S. Abraham, M. Raisee, G. Ghorbaniasl, F. Contino, C. Lacor, A robust and efficient stepwise regression method for building sparse polynomial chaos expansions, *J. Comput. Phys.* 332 (2017) 461–474.
- [26] S. Salehi, M. Raisee, M.J. Cervantes, A. Nourbakhsh, An efficient multifidelity l1-minimization method for sparse polynomial chaos, *Comput. Methods Appl. Mech. Eng.* 334 (2018) 183–207.
- [27] G.T. Buzzard, Global sensitivity analysis using sparse grid interpolation and polynomial chaos, *Reliab. Eng. Syst. Saf.* 107 (2012) 82–89.
- [28] P.G. Constantine, M.S. Eldred, E.T. Phipps, Sparse pseudospectral approximation method, *Comput. Methods Appl. Mech. Eng.* 229–232 (2012) 1–12.
- [29] J. Peng, J. Hampton, A. Doostan, A weighted l1-minimization approach for sparse polynomial chaos expansions, *J. Comput. Phys.* 267 (2014) 92–111.
- [30] J.D. Jakeman, A. Narayan, T. Zhou, A generalized sampling and preconditioning scheme for sparse approximation of polynomial chaos expansions, *SIAM J. Sci. Comp.* 39 (3) (2017) A1114–A1144.
- [31] K. Cheng, Z.Z. Lu, Sparse polynomial chaos expansion based on D-MORPH regression, *Appl. Math. Comput.* 323 (2018) 17–30.
- [32] K. Sargsyan, C. Safta, H.N. Najm, B.J. Debusschere, D. Ricciuto, P. Thornton, Dimensionality reduction for complex models via Bayesian compressive sensing, *Int. J. Uncertainty Quantif.* 4 (1) (2014) 63–93.
- [33] Y.C. Zhou, Z.Z. Lu, K. Cheng, Y. Shi, An expanded sparse Bayesian learning method for polynomial chaos expansion, *Mech. Syst. Sig. Process.* 128 (2019) 153–171.
- [34] J. Friedman, T. Hastie, H. Hofling, R. Tibshirani, Pathwise coordinate optimization, *Ann. Appl. Stat.* 1 (2) (2007) 302–332.
- [35] T.T. Wu, K. Lange, Coordinate descent algorithms for lasso penalized regression, *Ann. Appl. Statist.* 2 (1) (2008) 224–244.
- [36] J. Friedman, T. Hastie, R. Tibshirani, Regularization paths for generalized linear models via coordinate descent, *J. Stat. Softw.* 33 (1) (2010) 1–22.
- [37] S.J. Wright, Coordinate descent algorithms, *Math. Program.* 151 (1) (2015) 3–34.
- [38] R. Tibshirani, Regression shrinkage and selection via the lasso, *J. R. Statist. Soc. Series B-Statistical Methodol.* 58 (1) (1996) 267–288.
- [39] B. Efron, T. Hastie, I. Johnstone, R. Tibshirani, Least angle regression, *Ann. Statist.* 32 (2) (2004) 407–499.
- [40] Y.Y. Li, S. Osher, Coordinate descent optimization for l1-minimization with application to compressed sensing: a greedy algorithm, *Inverse Probl. Imag.* 3 (3) (2009) 487–503.
- [41] W. Yin, S. Osher, D. Goldfarb, J. Darbon, Bregman iterative algorithms for l1-minimization with applications to compressed sensing, *SIAM J. Imag. Sci.* 1 (1) (2008) 143–168.
- [42] D.L. Donoho, Compressed sensing, *IEEE Trans. Inf. Theory* 52 (4) (2006) 1289–1306.
- [43] J.A. Tropp, A.C. Gilbert, Signal recovery from random measurements via orthogonal matching pursuit, *IEEE Trans. Inf. Theory* 53 (12) (2007) 4655–4666.
- [44] Y.C. Eldar, G. Kutyniok, *Compressed Sensing: Theory and Applications*, Cambridge University Press, 2012.
- [45] Y.C. Zhou, Z.Z. Lu, K. Cheng, Sparse polynomial chaos expansions for global sensitivity analysis with partial least squares and distance correlation, *Struct. Multidiscip. Optim.* 59 (2019) 229–247.
- [46] H. Fang, Z.A. Fan, Y.F. Sun, M.P. Friedlander, Greed meets sparsity: Understanding and improving greedy coordinate descent for sparse optimization, in: *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, PMLR 108 (2020) 434–443.
- [47] I.M. Sobol', Sensitivity estimates for nonlinear mathematical models, *Mathem. Modeling and Computational Experiment* 1 (4) (1993) 407–414.
- [48] I.M. Sobol', Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates, *Mathem. Comp. Simul.* 55 (1–3) (2001) 271–280.
- [49] A. Saltelli, P. Annoni, I. Azzi, F. Campolongo, M. Ratto, S. Tarantola, Variance based sensitivity analysis of model output. Design and estimator for the total sensitivity index, *Comp. Phys. Commun.* 181 (2) (2010) 259–270.
- [50] T. Homma, A. Saltelli, Importance measures in global sensitivity analysis of nonlinear models, *Reliab. Eng. Syst. Saf.* 52 (1996) 1–17.
- [51] S.S. Chen, D.L. Donoho, M.A. Saunders, Atomic decomposition by basis pursuit, *SIAM J. Sci. Comp.* 20 (1) (1998) 33–61.
- [52] R. Baptista, V. Stolunov, P.B. Nair, Some greedy algorithms for sparse polynomial chaos expansions, *J. Comput. Phys.* 387 (2019) 303–325.
- [53] P. Tseng, Convergence of a block coordinate descent method for nondifferentiable minimization, *J. Optimiz. Theory Appl.* 109 (3) 475–494.
- [54] T. Hastie, R. Tibshirani, M. Wainwright, *Statistical Learning with Sparsity: The Lasso and Generalizations*, CRC Press, Boca Raton, 2015.
- [55] L.M. Bregman, The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming, *USSR Comput. Mathem. Mathem. Phys.* 7 (3) (1967) 200–217.
- [56] I.M. Sobol', On the distribution of points in a cube and the approximate evaluation of integrals, *USSR Comput. Mathem. Mathem. Phys.* 7 (4) (1967) 86–112.

- [57] T. Ishigami, T. Homma, An importance quantification technique in uncertainty analysis for computer models, in: *Proceedings of the First International Symposium on Uncertainty Modeling and Analysis (ISUMA'90)*, University of Maryland, 1990, pp. 398–403.
- [58] P. Zeng, *Fundamentals of Finite Element Analysis*, Higher Education Press, Beijing, 2009.
- [59] J. Xu, F. Kong, A cubature collocation based sparse polynomial chaos expansion for efficient structural reliability analysis, *Struct. Saf.* 74 (2018) 24–31.