Enhancing sparsity of Hermite polynomial expansions by iterative rotations

Xiu Yang [a,1], Huan Lei [a,1], Nathan A. Baker [b], Guang Lin [c,d,∗]

[a] *Advanced Computing, Mathematics, and Data Division, Pacific Northwest National Laboratory, Richland, WA 99352, USA*
[b] *Computational and Statistical Analytics Division, Pacific Northwest National Laboratory, Richland, WA 99352, USA*
[c] *Department of Mathematics, Purdue University, West Lafayette, IN 47907, USA*
[d] *School of Mechanical Engineering, Purdue University, West Lafayette, IN 47907, USA*

A R T I C L E   I N F O

A B S T R A C T

Compressive sensing has become a powerful addition to uncertainty quantification in recent years. This paper identifies new bases for random variables through linear mappings such that the representation of the quantity of interest is more sparse with new basis functions associated with the new random variables. This sparsity increases both the efficiency and accuracy of the compressive sensing-based uncertainty quantification method. Specifically, we consider rotation-based linear mappings which are determined iteratively for Hermite polynomial expansions. We demonstrate the effectiveness of the new method with applications in solving stochastic partial differential equations and high-dimensional ($\mathcal{O}(100)$) problems.

© 2015 Elsevier Inc. All rights reserved.

1. Introduction

Uncertainty quantification (UQ) plays an important role in constructing computational models as it helps to understand the influence of uncertainties on the quantity of interest. In this paper, we study parametric uncertainty, which treats some of the parameters as random variables. Let $(\Omega, \mathcal{F}, P)$ be a complete probability space, where Ω is the event space and P is a probability measure on the σ-field $\mathcal{F}$. We consider a system depending on a d-dimensional random vector $\boldsymbol{\xi}(\omega) = (\xi_1(\omega), \xi_2(\omega), \cdots, \xi_d(\omega))^T$, where ω is an event in Ω. For simplicity, we denote $\xi_i(\omega)$ as ξ_i. We aim to approximate the quantity of interest $u(\boldsymbol{\xi})$ with a generalized polynomial chaos (gPC) expansion [1,2]:

$$u(\boldsymbol{\xi}) = \sum_{n=1}^{N} c_n \psi_n(\boldsymbol{\xi}) + \varepsilon(\boldsymbol{\xi}), \tag{1.1}$$

where ε is the truncation error, N is a positive integer, c_n are coefficients, ψ_n are multivariate polynomials which are orthonormal with respect to the distribution of $\boldsymbol{\xi}$:

$$\int_{\mathbb{R}^d} \psi_i(\boldsymbol{\xi}) \psi_j(\boldsymbol{\xi}) \rho(\boldsymbol{\xi}) \mathrm{d}\boldsymbol{\xi} = \delta_{ij}, \tag{1.2}$$

* Corresponding author at: Department of Mathematics, Purdue University, West Lafayette, IN 47907, USA.
 E-mail address: guanglin@purdue.edu (G. Lin).
[1] The first two authors made equal contributions to this manuscript.

where $\rho(\boldsymbol{\xi})$ is the probability distribution function (PDF) of $\boldsymbol{\xi}$ and δ_{ij} is the Kronecker delta. The approximation converges in the L_2 sense as N increases if u is in the Hilbert space associated with the measure of $\boldsymbol{\xi}$ (i.e., the weight of the inner product is the PDF of $\boldsymbol{\xi}$) [2–4]. Stochastic Galerkin and probabilistic collocation are two popular methods [1,2,5–8] used to approximate the gPC coefficients $\boldsymbol{c} = (c_1, c_2, \cdots, c_N)^T$. Stochastic collocation starts by generating samples of input $\boldsymbol{\xi}^q, q = 1, 2, \cdots, M$ based on $\rho(\boldsymbol{\xi})$. Next, the computational model is calculated for each $\boldsymbol{\xi}^q$ to obtain corresponding samples of the output $u^q = u(\boldsymbol{\xi}^q)$. Finally, coefficients $\boldsymbol{c}$ are approximated based on u^q and $\boldsymbol{\xi}^q$. Note that in many practical problems, it is very costly to obtain u^q and, due to the limited computational sources, we will often have $M < N$ or even $M \ll N$. The smaller number of samples than basis functions implies that the following linear system is under-determined:

$$\boldsymbol{\Psi}\boldsymbol{c} = \boldsymbol{u} + \boldsymbol{\varepsilon}, \tag{1.3}$$

where $\boldsymbol{u} = (u^1, u^2, \cdots, u^M)^T$ is the vector of output samples, $\boldsymbol{\Psi}$ is an $M \times N$ matrix with $\Psi_{ij} = \psi_j(\boldsymbol{\xi}^i)$ and $\boldsymbol{\varepsilon} = (\varepsilon^1, \varepsilon^2, \cdots, \varepsilon^M)^T$ is a vector of error samples with $\varepsilon^q = \varepsilon(\boldsymbol{\xi}^q)$. The compressive sensing method is effective at solving this type of under-determined problem when $\boldsymbol{c}$ is sparse [9–12] and recent studies have applied this approach to uncertainty quantification (UQ) problems [13–24].

Several useful approaches have been developed to enhance the efficiency of solving Eq. (1.3) in UQ applications. First, re-weighted ℓ_1 minimization assigns a weight to each c_n and solves a weighted ℓ_1 minimization problem to enhance the sparsity [25]. The weights can be estimated in *a priori* [18,26] or, for more general cases, can be obtained iteratively [15, 17]. Second, better sampling strategies can be used, such as minimizing the mutual coherence [27,20]. Third, Bayesian compressive sensing method provides the posterior distribution of the coefficients [23,16]. Finally, adaptive basis selection selects basis functions to enhance the efficiency instead of fixing the basis functions at the beginning [22]. Recently, we propose an approach [17] to enhance the sparsity of $\boldsymbol{c}$ through the rotation of the random vector $\boldsymbol{\xi}$ to a new random vector $\boldsymbol{\eta}$, where the rotation operator is determined by the sorted variability directions of the quantity of interest u based on the active subspace method [28].

In this work, we aim to extend our previous work [17] and consider the specific case where the system depends on i.i.d. Gaussian random variables; i.e., $\boldsymbol{\xi} \sim \mathcal{N}(\boldsymbol{0}, I)$ where $\boldsymbol{0}$ is a d-dimensional zero vector and I is a $d \times d$ identity matrix. This assumption appears in a wide range of physics and engineering problems. We aim to find a mapping $g : \mathbb{R}^d \mapsto \mathbb{R}^d$ which maps $\boldsymbol{\xi}$ to a new set of i.i.d. Gaussian random variables $\boldsymbol{\eta} = (\eta_1, \eta_2, \cdots, \eta_d)^T$ such that the gPC expansion of u with respect to $\boldsymbol{\eta}$ is sparser. In other words,

$$u(\boldsymbol{\xi}) \approx \sum_{n=1}^{N} c_n \psi_n(\boldsymbol{\xi}) = \sum_{n=1}^{N} \tilde{c}_n \tilde{\psi}_n(\boldsymbol{\eta}(\boldsymbol{\xi})) \approx u(\boldsymbol{\eta}(\boldsymbol{\xi})), \tag{1.4}$$

where $\tilde{\psi}_n$ are orthonormal polynomials associated with the new random vector $\boldsymbol{\eta}$ and $\tilde{c}_n$ are the corresponding coefficients. Note that $\psi_n = \tilde{\psi}_n$ since $\boldsymbol{\eta} \sim \mathcal{N}(\boldsymbol{0}, I)$. We intend to find the set $\tilde{\boldsymbol{c}} = (\tilde{c}_1, \tilde{c}_2, \cdots, \tilde{c}_N)^T$ which is sparser than $\boldsymbol{c}$ while preserving the properties of matrix $\tilde{\boldsymbol{\Psi}}$ (with $\tilde{\Psi}_{ij} = \tilde{\psi}_j(\boldsymbol{\eta}^i)$) close to those of $\boldsymbol{\Psi}$ to improve the efficiency of the compressive sensing method. To accomplish this, we will use a linear mapping, based on the idea of active subspaces [28], to obtain $\boldsymbol{\eta}$ as first proposed in [17]. Unlike our previous work, we build this mapping iteratively in order to obtain a sparser $\tilde{\boldsymbol{c}}$ and improve the efficiency of the gPC approximation by compressive sensing. We also provide the analytical form of the "gradient matrix" (see Eq. (3.3)) to avoid estimating it with Monte Carlo methods. Our method is applicable for both ℓ_0 and ℓ_1 minimization problems. Especially, for the latter, we can also integrate the present method with re-weighted ℓ_1 minimization method to further reduce the error. We demonstrate that, compared with the standard compressive sensing methods, our approach reduces the relative L_2 error of the gPC approximation.

2. Brief review of the compressive sensing-based gPC method

2.1. Hermite polynomial chaos expansions

In this paper we study systems relying on d-dimensional Gaussian random vector $\boldsymbol{\xi} \sim \mathcal{N}(\boldsymbol{0}, I)$. Therefore, the gPC basis functions are constructed by tensor products of univariate orthonormal Hermite polynomials. For a multi-index $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \cdots, \alpha_d), \alpha_i \in \mathbb{N} \cup \{0\}$, we set

$$\psi_{\boldsymbol{\alpha}}(\boldsymbol{\xi}) = \psi_{\alpha_1}(\xi_1)\psi_{\alpha_2}(\xi_2) \cdots \psi_{\alpha_d}(\xi_d). \tag{2.1}$$

For two different multi-indices $\boldsymbol{\alpha}_i = ((\alpha_i)_1, (\alpha_i)_2, \cdots, (\alpha_i)_d)$ and $\boldsymbol{\alpha}_j = ((\alpha_j)_1, (\alpha_j)_2, \cdots, (\alpha_j)_d)$, we have the property

$$\int_{\mathbb{R}^d} \psi_{\boldsymbol{\alpha}_i}(\boldsymbol{\xi})\psi_{\boldsymbol{\alpha}_j}(\boldsymbol{\xi})\rho(\boldsymbol{\xi})\mathrm{d}\xi = \delta_{\boldsymbol{\alpha}_i\boldsymbol{\alpha}_j} = \delta_{(\alpha_i)_1(\alpha_j)_1}\delta_{(\alpha_i)_2(\alpha_j)_2} \cdots \delta_{(\alpha_i)_d(\alpha_j)_d}, \tag{2.2}$$

where

$$\rho(\boldsymbol{\xi}) = \left(\frac{1}{\sqrt{2\pi}}\right)^d \exp\left(-\frac{\xi_1^2 + \xi_2^2 + \cdots + \xi_d^2}{2}\right). \tag{2.3}$$

For simplicity, we denote $\psi_{\boldsymbol{\alpha}_i}(\boldsymbol{\xi})$ as $\psi_i(\boldsymbol{\xi})$.

2.2. Compressive sensing

The vector c in Eq. (1.3) can be approximated by solving the following optimization problem:

$$(P_{h,\epsilon}): \quad \arg\min_{\hat{c}} \|\hat{c}\|_h, \quad \text{subject to} \quad \|\Psi\hat{c} - u\|_2 \leq \epsilon, \tag{2.4}$$

where $\epsilon = \|\varepsilon\|_2$ and h is typically set as 0 or 1. For $h = 0$ (ℓ_0 minimization problem), the greedy Orthogonal Matching Pursuit (OMP) algorithm [29,12] can be applied; for $h = 1$ (ℓ_1 minimization problem), convex optimization methods are directly applicable [30]. As pointed out in [12], OMP is very efficient – when it works – but convergence to a sparse solution is not always guaranteed. There are specific cases where a sparse solution is possible while OMP yields a dense one. Since both the OMP and ℓ_1 minimization approaches are widely used, we will demonstrate the effectiveness of our new method for both methods.

Next, we introduce the concept of *sparsity* as it is critical in the error estimates for solving the under-determined system Eq. (1.3) with the compressive sensing method. The ℓ_0 "norm" of vector $x = (x_1, x_2, \cdots, x_N)$ is defined as the number of its non-zeros entries [31,9,12]

$$\|x\|_0 \overset{\text{def}}{=} \#\{i : x_i \neq 0\} \tag{2.5}$$

and ℓ_1 norm is defined as the sum of the absolute value of its entries:

$$\|x\|_1 \overset{\text{def}}{=} \sum_{n=1}^{N} |x_n|. \tag{2.6}$$

x is called *s-sparse* if $\|x\|_0 \leq s$, and x is considered a sparse vector if $s \ll N$. Few practical systems have a truly sparse gPC coefficients c. However, in many cases, the c are compressible, i.e., only a few entries make significant contribution to its ℓ_1 norm. In subsequent discussion, we relax the definition of "sparse": x is considered sparse if $\|x - x_s\|_1$ is small for $s \ll N$. Here x_s is defined as the **best s-sparse approximation** one could obtain if one knew exactly the locations and amplitudes of the s-largest entries of x, i.e., x_s is the vector x with all but the s-largest entries set to zero [11].

The error bound for solving Eq. (1.3) with ℓ_1 minimization requires definition of the *restricted isometry property* (RIP) constant [32]. For each integer $s = 1, 2, \cdots$, the isometry constant δ_s of a matrix Φ is defined as the smallest number such that

$$(1 - \delta_s)\|x\|_2^2 \leq \|\Phi x\|_2^2 \leq (1 + \delta_s)\|x\|_2^2 \tag{2.7}$$

holds for all s-sparse vectors x. With some restrictions, Candes et al. showed x can be stably reconstructed [11]. Assume that the matrix Ψ satisfies $\delta_{2s} < \sqrt{2} - 1$, and $\|\varepsilon\|_2 \leq \epsilon$, then solution $\hat{c}$ to $(P_{1,\epsilon})$ obeys

$$\|c - \hat{c}\|_2 \leq C_1\epsilon + C_2\frac{\|c - c_s\|_1}{\sqrt{s}}, \tag{2.8}$$

where C_1 and C_2 are constants, c is the exact vector we aim to approximate and $\hat{c}$ is the solution of $(P_{1,\epsilon})$. This result implies that the upper bound of the error is related to the truncation error and the sparsity of c, which is indicated in the first and second terms on the right hand side of Eq. (2.8), respectively.

The *re-weighted* ℓ_1 minimization approach is an improvement of the ℓ_1 minimization method, which enhances the accuracy of estimating c [25]. The re-weighted ℓ_1 approach solves the following optimization problem:

$$(P_{1,\epsilon}^W): \quad \arg\min_{\hat{c}} \|W\hat{c}\|_1, \text{subject to} \quad \|\Psi\hat{c} - u\|_2 \leq \epsilon, \tag{2.9}$$

where W is a diagonal matrix: $W = \text{diag}(w_1, w_2, \cdots, w_N)$. Clearly, $(P_{1,\epsilon})$ can be considered as a special case of $(P_{1,\epsilon}^W)$ by setting $W = I$. The elements w_i of the diagonal matrix can be estimated based on analysis of u as in Peng et al. [18], or be estimated iteratively [25,15]. More precisely, for each iteration l, $(P_{1,\epsilon}^W)$ is solved to obtain $\hat{c}^{(l)}$ and then $w_i^{(l+1)} = 1/(|\hat{c}_i^{(l)}| + \delta)$ for the next step. The parameter $\delta > 0$ is introduced to provide stability and to ensure that a zero-valued component in $\hat{c}^{(l)}$ does not prohibit a nonzero estimate at the next step. In Candes et al. [25], the authors suggest two to three iterations of this procedure. Subsequent analytical work [33] provides an error bound for each iteration as well as the limit of computing $\hat{c}$ with re-weighted ℓ_1 minimization. The form is similar to Eq. (2.8) with different constants.

In practice, the error term ϵ is not known *a priori*, hence cross-validation is needed to estimate it. One such algorithm is [13] summarized in Algorithm 1.

We omit the review of the theoretical results for the OMP as well as its variants, and refer interested readers to the literature [12,34,35]. Similar to the ℓ_1 approach, the error estimate for OMP includes a term which depends on the sparsity of c. This is a critical point that motivates us to propose the new method described in the next section.

Algorithm 1 Cross-validation to estimate the error ϵ.

1: Divide the M output samples to M_r reconstruction ($\boldsymbol{u}_r$) and M_v validation ($\boldsymbol{u}_v$) samples and divide the measurement matrix $\boldsymbol{\Psi}$ correspondingly into $\boldsymbol{\Psi}_r$ and $\boldsymbol{\Psi}_v$.
2: Choose multiple values for ϵ_r such that the exact error $\|\boldsymbol{\Psi}_r \boldsymbol{c} - \boldsymbol{u}_r\|_2$ of the reconstruction samples is within the range of ϵ_r values.
3: For each ϵ_r, solve $(P_{h,\epsilon})$ with $\boldsymbol{u}_r$ and $\boldsymbol{\Psi}_r$ to obtain $\hat{\boldsymbol{c}}$, then compute $\epsilon_v = \|\boldsymbol{\Psi}_v \hat{\boldsymbol{c}} - \boldsymbol{u}_v\|_2$.
4: Find the minimum value of ϵ_v and its corresponding ϵ_r. Set $\epsilon = \sqrt{M/M_r}\epsilon_r$.

2.3. Compressive sensing-based gPC methods

Given M samples of $\boldsymbol{\xi}$, the quantity of interest u is approximated by a gPC expansion as in Eq. (1.1):

$$u(\boldsymbol{\xi}^q) = \sum_{n=1}^{N} c_n \psi(\boldsymbol{\xi}^q) + \varepsilon(\boldsymbol{\xi}^q), \quad q = 1, 2, \cdots, M, \tag{2.10}$$

which can be rewritten as Eq. (1.3). A typical approach to compressive sensing based-gPC is summarized in Algorithm 2.

Algorithm 2 Compressive sensing-based gPC.

1: Generate input samples $\boldsymbol{\xi}^q, q = 1, 2, \cdots, M$ based on the distribution of $\boldsymbol{\xi}$.
2: Generate output samples $u^q = u(\boldsymbol{\xi}^q)$ by solving the complete model; e.g., running simulations, solvers, etc.
3: Select gPC basis functions $\{\psi_n\}_{n=1}^{N}$ associated with $\boldsymbol{\xi}$ and then generate the measurement matrix $\boldsymbol{\Psi}$ by setting $\Psi_{ij} = \psi_j(\boldsymbol{\xi}^i)$.
4: Solve the optimization problem $(P_{h,\epsilon})$:

$$\arg\min_{\hat{\boldsymbol{c}}} \|\hat{\boldsymbol{c}}\|_h, \text{ subject to } \|\boldsymbol{\Psi}\hat{\boldsymbol{c}} - \boldsymbol{u}\|_2 \le \epsilon,$$

where $h = 0$ or 1, $\boldsymbol{u} = (u^1, u^2, \cdots, u^M)^T$, and ϵ is obtained by cross-validation. If the re-weighted ℓ_1 method is employed, solve $(P_{1,\epsilon}^W)$ instead.
5: Set $\boldsymbol{c} = \hat{\boldsymbol{c}}$ and construct gPC expansion as $u(\boldsymbol{\xi}) \approx \sum_{n=1}^{N} c_n \psi_n(\boldsymbol{\xi})$.

Note that the RIP condition in Theorem 2.2 is sufficient but not necessary; furthermore, it is difficult to obtain the exact RIP constant in practical problems. A more tractable property of the measurement matrix for calculation is the *mutual coherence* [12]:

$$\mu(\boldsymbol{\Psi}) = \max_{1 \le j,k \le N, j \ne k} \frac{|\boldsymbol{\Psi}_j^T \boldsymbol{\Psi}_k|}{\|\boldsymbol{\Psi}_j\|_2 \cdot \|\boldsymbol{\Psi}_k\|_2}, \tag{2.11}$$

where $\boldsymbol{\Psi}_j$ and $\boldsymbol{\Psi}_k$ are columns of $\boldsymbol{\Psi}$. In general, a measurement matrix with smaller mutual coherence is better able to recover a sparse solution with the compressive sensing method. Note that $\mathbb{E}\left\{\psi_i(\xi)\psi_j(\xi)\right\} = \delta_{ij}$ since $\{\psi_i\}_{i=1}^{N}$ are orthonormal polynomials. Therefore, asymptotically, $\mu(\boldsymbol{\Psi})$ converges to zeros according to the strong law of large numbers.

In the next section, we will demonstrate that our new method increases the sparsity of $\boldsymbol{c}$ without changing μ significantly, and hence, our method is able to improve the accuracy of the compressive sensing-based gPC method.

3. Iterative rotations for increasing sparsity

In this section, we provide a heuristic method to identify the rotation matrix by computing the eigenvalue decomposition of a gradient matrix G. The rotation increases the sparsity of the gPC expansions of quantity of interest u with respect to a new set of random variables. The rotation procedure is applied iteratively to achieve a target sparsity level. The enhancement of the sparsity decreases the second term (sparsity-induced error) on the right hand side of Eq. (2.8). For the cases where this sparsity-induced error dominates the total error of the compressive sensing method, our new approach improves the overall accuracy.

From Eq. (2.8), we notice that if $\boldsymbol{c}$ is exactly sparse (i.e., $\boldsymbol{c} = \boldsymbol{c}_{s^*}$ for some $s^* \ll N$) and if the RIP condition is satisfied for $s \ge s^*$, then $\|\boldsymbol{c} - \boldsymbol{c}_s\| = 0$. Therefore, the upper bound of the error only depends on ϵ. In practical problems, $\boldsymbol{c}$ is usually not exactly sparse. But if the truncation error ϵ is sufficiently small, then the second term on the right hand side of Eq. (2.8) dominates the upper bound of the error. Hence, in order to improve the accuracy of the gPC expansion, we need to decrease $\|\boldsymbol{c} - \boldsymbol{c}_s\|_1/\sqrt{s}$. However, once the gPC basis functions are selected, $\boldsymbol{c}$, and therefore $\|\boldsymbol{c} - \boldsymbol{c}_s\|_1/\sqrt{s}$, are fixed. A natural way to enhance the sparsity of $\boldsymbol{c}$ is to find another set of random variables $\boldsymbol{\eta} = (\eta_1, \eta_2, \cdots, \eta_{\tilde{d}})^T$, which depend on $\boldsymbol{\xi}$ such that the vector $\tilde{\boldsymbol{c}}$, which are the gPC coefficients of u with respect to $\boldsymbol{\eta}$, is sparser. In order words, our goal is to seek $\boldsymbol{\eta}(\boldsymbol{\xi})$ with

$$u(\boldsymbol{\xi}) \approx \sum_{n=1}^{N} c_n \psi_n(\boldsymbol{\xi}) = \sum_{n=1}^{\tilde{N}} \tilde{c}_n \tilde{\psi}_n(\boldsymbol{\eta}(\boldsymbol{\xi})) \approx u(\boldsymbol{\eta}(\boldsymbol{\xi})),$$

such that $\|\tilde{\boldsymbol{c}} - \tilde{\boldsymbol{c}}_s\|_1 < \|\boldsymbol{c} - \boldsymbol{c}_s\|_1$. Note that $\tilde{N}$ does not necessarily equal N and $\tilde{d}$ can be different from d. We will denote the mapping from $\boldsymbol{\xi}$ to $\boldsymbol{\eta}$ as $g : \mathbb{R}^d \mapsto \mathbb{R}^{\tilde{d}}$.

There are several behaviors that our proposed approach must exhibit.

- *The PDF of $\boldsymbol{\eta}$ must be computed efficiently.* The first step of generating a new gPC expansion is to obtain the PDF of $\boldsymbol{\eta}$. Hence, if g is complicated, the PDF of $\boldsymbol{\eta}$ will be difficult to obtain. Even if g is a simple function, it can still be difficult to obtain an accurate PDF if the dimension is large.
- *The new gPC basis functions associated with $\boldsymbol{\eta}$ must be computed efficiently.* If the η_i are independent, then the new gPC basis can be constructed as the tensor product of univariate basis functions in each dimension. Although this is not necessary, it will make the construction of new basis functions easier as it avoids the computation of high-dimensional integrals.
- *The properties of the measurement matrix must be preserved.* Clearly, the measurement matrix changes as we introduce new random variables and new basis functions. Even though we may construct a very sparse $\tilde{\boldsymbol{c}}$, if the key properties of the measurement matrix are altered too much (e.g., the RIP constant or mutual coherence increases dramatically), we may be unable to obtain an accurate result with the compressive sensing method.
- *No additional output samples must be needed.* In particular, the existing output samples $u^q, q = 1, 2, \cdots, M$ should be sufficient. This is especially important for the cases when the model (simulation or deterministic solver) is very costly to compute.

In this work, we focus on the special case of normal distributions $\boldsymbol{\xi} \sim \mathcal{N}(\mathbf{0}, I)$; hence, the ψ_i are constructed as the tensor product of univariate orthonormal Hermite polynomials as shown in Eq. (2.1). We aim to find a linear mapping $g : \mathbb{R}^d \mapsto \mathbb{R}^d$ such that

$$\boldsymbol{\eta} = g(\boldsymbol{\xi}) = A\boldsymbol{\xi}, \tag{3.1}$$

where A is an orthonormal matrix.

If we find this matrix A then all of the aforementioned behaviors can be obtained. We know that $\boldsymbol{\eta} \sim \mathcal{N}(\mathbf{0}, I)$ since $AA^T = I$. Therefore, the samples of $\boldsymbol{\eta}$ can be obtained as $\boldsymbol{\eta}^q = A\boldsymbol{\xi}^q$, where $\boldsymbol{\xi}^q$ are generated at the beginning (Step 1 in Algorithm 2). Since $\boldsymbol{\eta} \sim \mathcal{N}(\mathbf{0}, I)$ we can set $\tilde{\psi}_i = \psi_i$ and no additional computation is needed. The difference between Ψ and $\tilde{\Psi}$ is that the latter is constructed by evaluating orthonormal Hermite polynomials at another set of samples of i.i.d. Gaussian random variables; i.e., $\tilde{\Psi}_{ij} = \tilde{\psi}_j(\boldsymbol{\eta}^i) = \psi_j(\boldsymbol{\eta}^i)$. Therefore, the mutual coherence of $\tilde{\Psi}$ converges to 0 as that of Ψ, and the difference between $\mu(\Psi)$ and $\mu(\tilde{\Psi})$ is $\mathcal{O}(M^{-1/2})$, the deviation of the Monte Carlo numerical integral from the exact value. No additional samples u^q are required since the improvement of accuracy is achieved by enhancing the sparsity of gPC coefficients.

Given the Hermite polynomials defined above, we have a new expansion for u:

$$u(\boldsymbol{\xi}) \approx \sum_{n=1}^{N} c_n \psi_n(\boldsymbol{\xi}) = \sum_{n=1}^{N} \tilde{c}_n \psi_n(A\boldsymbol{\xi}) \approx u(\boldsymbol{\eta}) \tag{3.2}$$

with $\tilde{\boldsymbol{c}}$ sparser than $\boldsymbol{c}$. In order to obtain the A, we adopt the active subspace approach [28]. We first define the "gradient matrix":

$$G \stackrel{\text{def}}{=} \mathbb{E}\left\{\nabla u(\boldsymbol{\xi}) \cdot \nabla u(\boldsymbol{\xi})^T\right\} = U\Lambda U^T, \quad UU^T = I, \tag{3.3}$$

where G is symmetric, $\nabla u(\boldsymbol{\xi}) = (\partial u/\partial \xi_1, \partial u/\partial \xi_2, \cdots, \partial u/\partial \xi_d)^T$ is a column vector, $U = (\boldsymbol{U}_1, \boldsymbol{U}_2, \cdots, \boldsymbol{U}_d)$ is an orthonormal matrix consisting of eigenvectors $\boldsymbol{U}_i$, and $\Lambda = \text{diag}(\lambda_1, \lambda_2, \cdots, \lambda_d)$ with $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d$ is a diagonal matrix with elements representing decreasing variation of the system along the respective eigenvectors. We choose $A = U^T$ which, as a unitary matrix, defines a rotation in $\mathbb{R}^d$ and the linear mapping g in Eq. (3.1) projects $\boldsymbol{\xi}$ on the eigenvectors $\boldsymbol{U}_i$. Consequently, when the differences between $|\lambda_i|$ are large, g helps to concentrate the dependence of u primarily on the first few new random variables η_i due to the larger variation of u along the directions of the corresponding eigenvectors. Therefore, we obtain a sparser $\tilde{\boldsymbol{c}}$ than $\boldsymbol{c}$. We note that this approach of constructing G is similar to the method of outer product gradients (OPGs) in statistics [36,37]. The information of the gradient of u is also utilized to improve the efficiency of compressive sensing in the *gradient-enhanced* method [38–40,22,24].

Since u is not known *a priori*, we replace it with its gPC expansion $u_g = \sum_{n=1}^{N} c_n \psi_n(\boldsymbol{\xi})$. In prior work by Constantine and others [28,17], the expectation is obtained by taking the average of the Monte Carlo results. In the current work, we compute G differently: after obtaining $\boldsymbol{c}$ with compressive sensing method, we construct a gPC approximation u_g to u and approximate G accordingly:

$$G \approx \mathbb{E}\left\{\nabla\left(\sum_{n=1}^{N} c_n \psi_n(\boldsymbol{\xi})\right) \cdot \nabla\left(\sum_{n'=1}^{N} c_{n'} \psi_{n'}(\boldsymbol{\xi})\right)^T\right\}. \tag{3.4}$$

The entries of G can be approximated as:

$$G_{ij} \approx \mathbb{E}\left\{ \frac{\partial}{\partial \xi_i}\left(\sum_{n=1}^{N} c_n \psi_n(\boldsymbol{\xi})\right) \cdot \frac{\partial}{\partial \xi_j}\left(\sum_{n'=1}^{N} c_{n'} \psi_{n'}(\boldsymbol{\xi})\right)\right\}$$

$$= \mathbb{E}\left\{\left(\sum_{n=1}^{N} c_n \frac{\partial \psi_n(\boldsymbol{\xi})}{\partial \xi_i}\right) \cdot \left(\sum_{n'=1}^{N} c_{n'} \frac{\partial \psi_{n'}(\boldsymbol{\xi})}{\partial \xi_j}\right)\right\}$$

$$= \sum_{n=1}^{N} \sum_{n'=1}^{N} c_n c_{n'} \mathbb{E}\left\{\frac{\partial \psi_n(\boldsymbol{\xi})}{\partial \xi_i} \cdot \frac{\partial \psi_{n'}(\boldsymbol{\xi})}{\partial \xi_j}\right\}$$

$$= \boldsymbol{c}^T K_{ij} \boldsymbol{c}, \tag{3.5}$$

where K_{ij} is a "stiffness" matrix with entries

$$(K_{ij})_{kl} = \mathbb{E}\left\{\frac{\partial \psi_k(\boldsymbol{\xi})}{\partial \xi_i} \cdot \frac{\partial \psi_l(\boldsymbol{\xi})}{\partial \xi_j}\right\}. \tag{3.6}$$

Notice that K_{ij} can be precomputed since $\{\psi_i\}$ are normalized Hermite polynomials (see Appendix for details). G is a $d \times d$ matrix, where d is the number of random variables in the system. Since G is a symmetric matrix, we only need to compute $d(d+1)/2$ of its entries. Furthermore, unlike the active subspace method, which focuses on the *subspace* of $\mathbb{R}^d$, we keep the dimension and set of basis functions unchanged.

The entire iterative procedure is summarized in Algorithm 3. This algorithm adds post-processing steps to Algorithm 2,

---

Algorithm 3 Compressive sensing method with iterative rotations.

1: For given random vector $\boldsymbol{\xi}$ and quantity of interest u, run Algorithm 2 to obtain approximated gPC coefficients $\hat{\boldsymbol{c}}$.
2: Set counter $l = 0$, $\eta^{(0)} = \boldsymbol{\xi}$, $\tilde{\boldsymbol{c}}^{(0)} = \hat{\boldsymbol{c}}$.
3: Construct G^{l+1} with $\hat{\boldsymbol{c}}^{(l)}$ according to Eq. (3.5). Then decompose $G^{(l+1)}$ as

$$G^{(l+1)} = U^{(l+1)} \Lambda^{(l+1)} (U^{(l+1)})^T, \quad U^{(l+1)}(U^{(l+1)})^T = I.$$

4: Define $\boldsymbol{\eta}^{(l+1)} = (U^{(l+1)})^T \boldsymbol{\eta}^{(l)}$, and compute samples $(\boldsymbol{\eta}^{(l+1)})^q = (U^{(l+1)})^T (\boldsymbol{\eta}^{(l)})^q, q = 1, 2, \cdots, M$. Also, construct the new measurement matrix $\boldsymbol{\Psi}^{(l+1)}$ with $\Psi_{ij}^{(l+1)} = \psi_j((\boldsymbol{\eta}^{(l+1)})^i)$.
5: Solve the optimization problem $(P_{h,\epsilon^{(l+1)}})$:

$$\arg\min_{\hat{\boldsymbol{c}}} \|\hat{\boldsymbol{c}}\|_h, \quad \text{subject to } \|\boldsymbol{\Psi}^{(l+1)}\hat{\boldsymbol{c}} - \boldsymbol{u}\|_2 \le \epsilon^{(l+1)},$$

and set $\tilde{\boldsymbol{c}}^{(l+1)} = \hat{\boldsymbol{c}}$. If reweight ℓ_1 method is employed, solve $(P_{1,\epsilon^{(l+1)}}^W)$ instead.
6: Set $l = l + 1$. If $\left| \|U^{(l)}\|_1 - d\right| < \theta$, where the threshold θ is a positive real number, then stop. Otherwise, go to Step 3.

---

which is designed to increase the accuracy of compressive sensing based gPC method. In Step 5, we use notation $\epsilon^{(l+1)}$ since the estimated error at iteration $l+1$ may be different from ϵ. According to our numerical experiments (see Sec. 4), it is usually sufficient to test two or three different values on $[\epsilon/5, \epsilon]$ in the cross-validation procedure (see Algorithm 1) to obtain $\epsilon^{(l+1)}$.

In Algorithm 3, we propose a terminating condition based on the ℓ_1 norm of the rotation matrix in each iteration: $S(U^{(l)}) \overset{\text{def}}{=} \sum_{i=1}^{d} \|\boldsymbol{U}_i^{(l)}\|_1$. If $U^{(l)}$ is the identity matrix or a permutation matrix, we need no further iterations, and $S(U^{(l)}) = d$. Otherwise, $S(U^{(l)}) > d$ since $\|\boldsymbol{U}_i^{(l)}\|_2 = 1$ and

$$\left\|\boldsymbol{U}_i^{(l)}\right\|_1^2 = \left(\sum_{j=1}^{d}\left|(U_i^{(l)})_j\right|\right)^2 = \left\|\boldsymbol{U}_i^{(l)}\right\|_2^2 + 2\sum_{1 \le j, k \le d, j \ne k}\left|(U_i^{(l)})_j\right|\left|(U_i^{(l)})_k\right| > 1. \tag{3.7}$$

Hence, one may set a threshold θ and the iteration stops when $|S(U^{(l)}) - d| < \theta$. Empirically, θ can be set around $0.1d$–$0.2d$. More sophisticated terminating conditions (e.g., sparsity or ϵ estimates) are also possible. A rigorous theoretical analysis on the convergence behavior is not available at this time. The criterion presented here provides an approach to estimate, to some extend, whether our method converges. Empirically, when this stopping criterion is satisfied, additional iterations will not improve the accuracy significantly. We also note that the simplest terminating condition in Step 6 is to set a maximum iteration steps L. Based on our numerical examples in Sec. 4, this simple condition can also be useful. In general, the efficiency of our method depends on the intrinsic sparsity of the system, i.e., whether the system mainly relies on a small amount of subspaces. The fewer subspaces the system depends on, the better performance our method exhibits. Otherwise, this method is less effective, e.g., an extreme case is $u(\boldsymbol{\xi}) = \sum_{i=1}^{d} \xi_i^2$, for which the iterative rotations based on current framework does not help.

4. Numerical results

In this section, we present five numerical examples to demonstrate the effectiveness of our new method. The accuracies of different methods are measured by the relative L_2 error: $(\|u - u_g\|_2)/\|u\|_2$, where u_g is the Hermite polynomial expansion of u. The integral

$$\|u(\boldsymbol{\xi})\|_2 = \left(\int_{\mathbb{R}^d} u(\boldsymbol{\xi})^2 \rho(\boldsymbol{\xi}) \mathrm{d}\boldsymbol{\xi} \right)^{1/2} \tag{4.1}$$

(and $\|u - u_g\|_2$) is approximated with a high-level sparse grid method which is based on one-dimensional Gauss–Hermite quadrature and the Smolyak structure [41]. The term "level" p means that the algebraic accuracy of the sparse grid method is $2p - 1$. We use P to denote the truncation order, which implies that Hermite polynomials up to order P are included in expansion u_g. Hence, the number of unknowns can be computed as $N = \binom{P+d}{d}$.

The relative errors we present in this section are obtained from 100 independent replicates for each sample size M. For example, we generate 100 independent sets of input samples $\boldsymbol{\xi}^q, q = 1, 2, \cdots, M$, compute 100 different relative errors, and then report the average of these error samples. To investigate the effectiveness of the increasing of output samples, we set the x-axis in our figures as the ratio M/N which is the fraction of available data with respect to number of unknowns. We use MATLAB package `SPGL1` [42,43] to solve ($P_{1,\epsilon}$) as well as ($P_{1,\epsilon}^W$) and use `SparseLab` [44] for the OMP method. If not otherwise indicated, results are obtained with $L = 3$ iterations in Step 6 of Algorithm 3.

4.1. Example function with equally important random variables

Consider the following function

$$u(\boldsymbol{\xi}) = \sum_{i=1}^{d} \xi_i + 0.25 \left(\sum_{i=1}^{d} \xi_i \right)^2 + 0.025 \left(\sum_{i=1}^{d} \xi_i \right)^3, \tag{4.2}$$

where all ξ_i are equally important. In this case, adaptive methods that build the surrogate model hierarchically based on the importance of ξ_i (e.g., [45–48]) may not be efficient. A simple rotation matrix for this example has the form

$$A = \begin{pmatrix} d^{-1/2} & d^{-1/2} & \cdots & d^{-1/2} \\ & \tilde{A} & & \end{pmatrix},$$

where $\tilde{A}$ is a $(d-1) \times d$ matrix chosen to ensure that A is orthonormal. Given this choice for A, then $\eta_1 = (\sum_{i=1}^{d} \xi_i)/d^{1/2}$ and u has a very simple representation:

$$u(\boldsymbol{\xi}) = u(\boldsymbol{\eta}) = d^{1/2} \eta_1 + 0.25 d \eta_1^2 + 0.025 d^{3/2} \eta_1^3.$$

Therefore, as we keep the set of the basis functions unchanged, all the Hermite polynomials not related to η_1 make no contribution to the expansion, which implies that we obtain a very sparse representation of u. Unfortunately, the optimal structure is not known *a priori*, hence, the standard compressive sensing cannot take advantage of it.

In this test, we set $d = 12$ (hence, $N = 455$ for $P = 3$) and demonstrate the effectiveness of our new method. The integrals for calculating the L_2 error are computed by a level 4 sparse grid method, hence they are exact. The relative error of ℓ_1 minimization and OMP are presented in Fig. 1. Clearly, the standard ℓ_1 minimization and OMP are not effective as the relative error is close to 100% even when $M/N > 0.4$. Also, the re-weighted ℓ_1 does not help in this case. However, our new iterative rotation demonstrates much better accuracy, especially when M is large. As demonstrated in Fig. 2 the iterative rotation creates a much sparser representation of u, hence the efficiency of compressive sensing method is substantially enhanced. We notice that the accuracy increases as more iterations are included. However, the improvement from 6 iterations to 9 iterations is less significant as that from 3 iterations to 6 iterations, especially for the OMP-based iterative rotation method. In general, the improvement afforded by iterative rotation becomes small after 3 iterations.

This contrived example demonstrates that our new method is capable of enhancing the sparsity of the Hermite polynomial expansion, even with a very inaccurate $\tilde{\boldsymbol{c}}^{(0)}$ in Step 2 of Algorithm 3 when other methods fail.

4.2. Example function with high compressibility

Consider the following function:

$$u(\boldsymbol{\xi}) = \sum_{|\boldsymbol{\alpha}|=0}^{P} c_{\boldsymbol{\alpha}} \psi_{\boldsymbol{\alpha}}(\boldsymbol{\xi}) = \sum_{n=1}^{N} c_n \psi_n(\boldsymbol{\xi}), \quad \boldsymbol{\xi} = (\xi_1, \xi_2, \cdots, \xi_d), \tag{4.3}$$
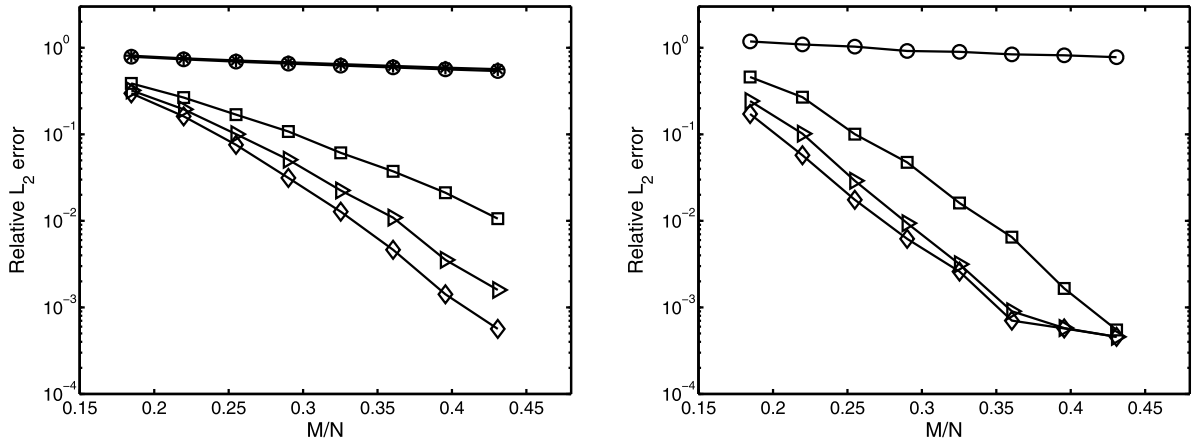
Fig. 1. Results for the example function with equally important random variables. (Left) Comparison with the ℓ_1 method. "○": ℓ_1, "□": rotated ℓ_1 with 3 iterations, "▷": rotated ℓ_1 with 6 iterations, "◇": rotated ℓ_1 with 9 iterations, "∗": re-weighted ℓ_1. (Right) Comparison with the OMP method. "○": OMP, "□": rotated OMP with 3 iterations, "▷": rotated OMP with 6 iterations, "◇": rotated OMP with 9 iterations. These calculations were performed with dimension $d = 12$ and the number of unknowns $N = 455$.
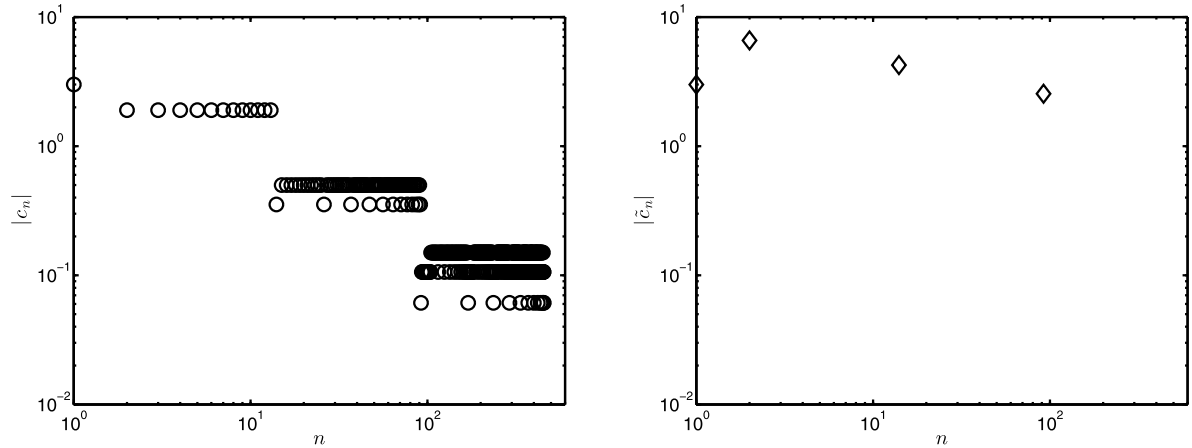
Fig. 2. Magnitude of the gPC coefficients for the example function with equally important random variables. (Left) Magnitude of c_n. (Right) Magnitude of $\tilde{c}_n$ of a randomly chosen replicate computed by rotated ℓ_1 with 9 iterations and $M = 180$ ($M/N \approx 0.4$). These calculations were performed with dimension $d = 12$ and the number of unknowns $N = 455$.

where, ψ_α are normalized multivariate Hermite polynomials, $d = 12$, $P = 3$, $N = 455$, and the coefficients c_n are chose as uniformly distributed random numbers,

$$c_n = \zeta / n^{1.5}, \quad \zeta \sim \mathcal{U}[-1, 1]. \tag{4.4}$$

For this example, we generate N samples of ζ: $\zeta^1, \zeta^2, \cdots, \zeta^N$ then divide them by $n^{1.5}, n = 1, 2, \cdots, N$ to obtain a random "compressible signal" c. The integrals for the relative error are computed by a level-4 sparse grid method and are therefore exact. Fig. 3 shows the relative error with different numbers of iterations (1–3) for the ℓ_1 minimization and OMP methods. Our new iterative rotation method improves the accuracy of the gPC approximation for both methods. As before, benefit of increased iterations drops sharply near $L = 3$. Therefore, in the remainder of this paper we use $L = 3$ iterations unless otherwise noted.

Fig. 4 shows results obtained by applying our iterative rotation technique to the re-weighted ℓ_1 approach using $L = 3$ iterations. The results for the iterative rotation approach for OMP are also presented in Fig. 4. For all methods, introduction of the iterative rotation approach improves the results. A comparison of the sparsity of c and $\tilde{c}$ is presented in Fig. 5. The main improvement is that the number of coefficients with magnitude larger than 0.01 is decreased. Also, $|c_n|$ cluster around the line $|c_n| = 1/n^{1.5}$ as we set them in this way, while many $|\tilde{c}_n|$ are much below this line especially when n is large.
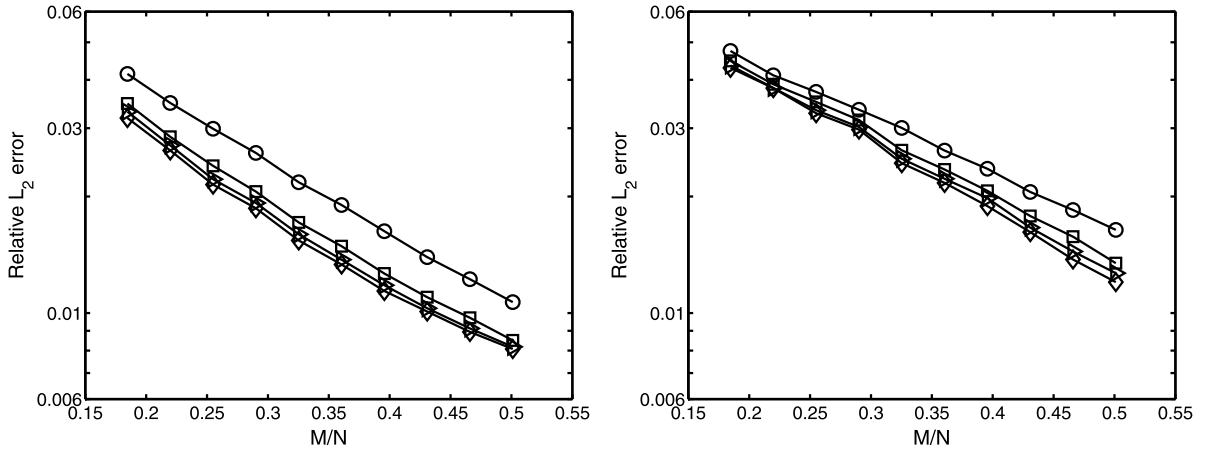
Fig. 3. Results for the example function with high compressibility. Relative L_2 error for different numbers of iterations for ℓ_1 minimization (left) and the OMP method (right). "○": standard ℓ_1 (left) or standard OMP (right), "□": 1 iteration, "▷": 2 iterations, "◇": 3 iterations. These calculations were performed with dimension $d = 12$ and number of unknowns $N = 455$.
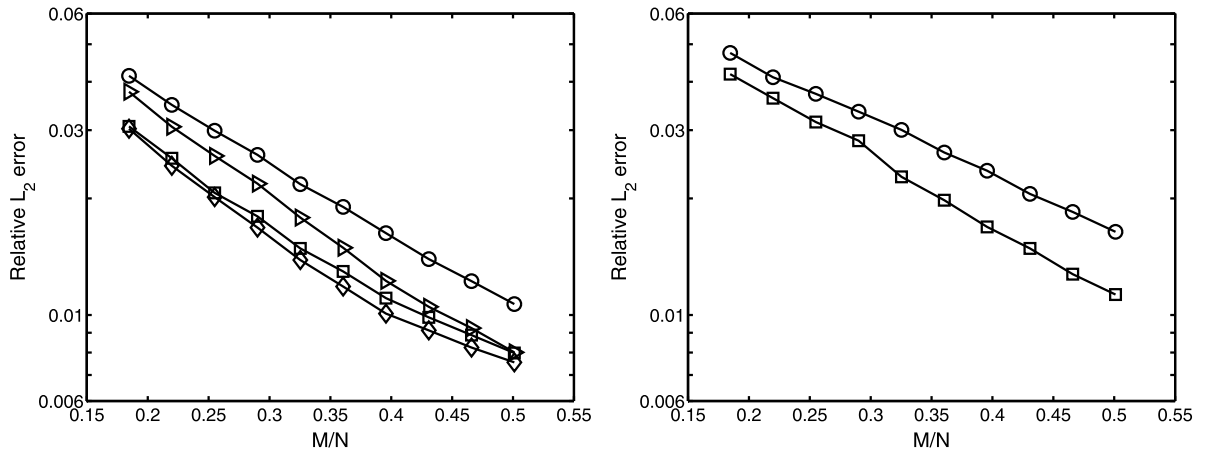

Fig. 4. Results for the example function with high compressibility. (Left) Comparison with ℓ_1 methods. "○": standard ℓ_1, "▷": re-weighted ℓ_1, "□": rotated ℓ_1, "◇": re-weighted and iteratively rotated ℓ_1. (Right) Comparison with OMP methods. "○": OMP, "□": rotated OMP. These calculations were performed with dimension $d = 12$ and number of unknowns $N = 455$.
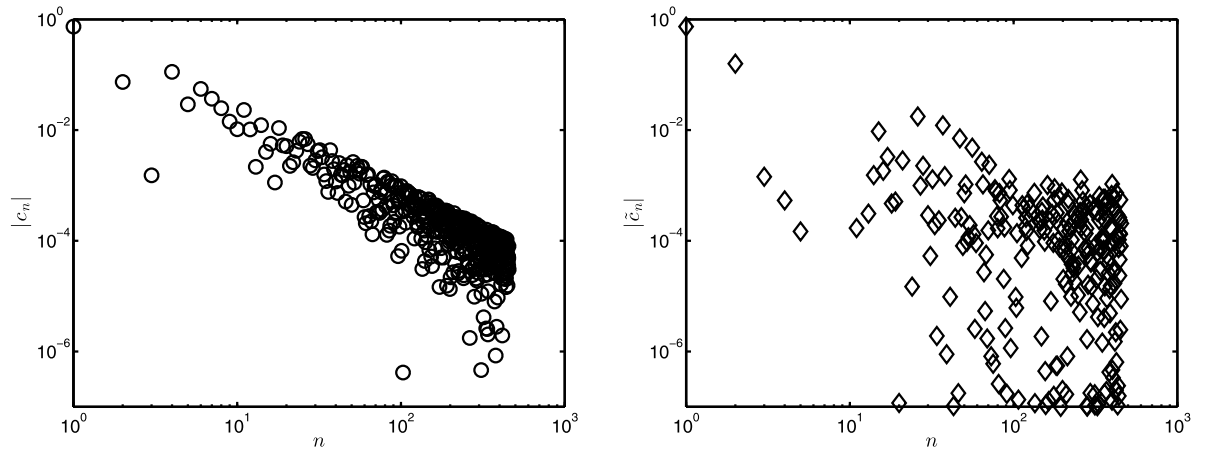
Fig. 5. Magnitude of the gPC coefficients for example function with high compressibility. (Left) Magnitude of $\boldsymbol{c}_n$. (Right) Magnitude of $\tilde{\boldsymbol{c}}_n$ of a randomly chosen replicate computed by re-weighted and iteratively rotated ℓ_1 with $M = 180$ ($M/N \approx 0.4$). These calculations were performed with dimension $d = 12$ and the number of unknowns $N = 455$.
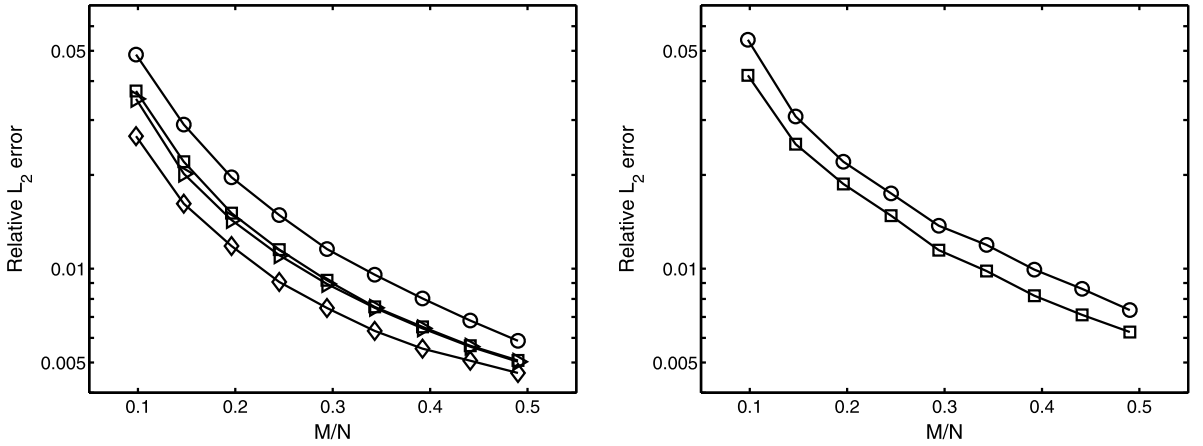
Fig. 6. Results for the example elliptic differential equation. (Left) Comparison with ℓ_1 methods. "∘": standard ℓ_1, "▷": re-weighted ℓ_1, "□": rotated ℓ_1, "◇": re-weighted and iteratively rotated ℓ_1. (Right) Comparison with OMP methods. "∘": standard OMP, "□": rotated OMP. These calculations were performed with dimension $d = 15$ and the number of unknowns $N = 816$.

4.3. Example elliptic differential equation

Next we consider a one-dimensional elliptic differential equation with a random high-order coefficient:

$$-\frac{d}{dx}\left(a(x;\boldsymbol{\xi})\frac{du(x;\boldsymbol{\xi})}{dx}\right) = 1, \quad x \in (0,1)$$
$$u(0) = u(1) = 0, \tag{4.5}$$

where $a(x;\boldsymbol{\xi})$ is a log-normal random field based on Karhunen–Loève (KL) expansion:

$$a(x;\boldsymbol{\xi}) = a_0(x) + \exp\left(\sigma\sum_{i=1}^{d}\sqrt{\lambda_i}\phi_i(x)\xi_i\right), \tag{4.6}$$

where $\{\xi_i\}$ are i.i.d. standard Gaussian random variables, $\{\lambda_i\}_{i=1}^{d}$, and $\{\phi_i(x)\}_{i=1}^{d}$ are the largest eigenvalues and corresponding eigenfunctions of the exponential covariance kernel:

$$C(x,x') = \exp\left(\frac{|x-x'|}{l_c}\right). \tag{4.7}$$

In the KL expansion, λ_i denotes the eigenvalue of the covariance kernel $C(x,x')$ instead of entries of Λ in Eq. (3.5). The value of λ_i and the analytical expressions for ϕ_i are available in the literature [49]. In this example, we set $a_0(x) \equiv 0.1, \sigma = 0.5, l_c = 0.2$ and $d = 15$. With this setting, $\sum_{i=1}^{d}\lambda_i > 0.93\sum_{i=1}^{\infty}\lambda_i$. For each input sample $\boldsymbol{\xi}^q$, a and u only depend on x and the solution of the deterministic elliptic equation can be obtained as [15]:

$$u(x) = u(0) + \int_0^x \frac{a(0)u(0)' - y}{a(y)}dy. \tag{4.8}$$

By imposing the boundary condition $u(0) = u(1) = 0$, we can compute $a(0)u(0)'$ as

$$a(0)u(0)' = \left(\int_0^1 \frac{y}{a(y)}dy\right) / \left(\int_0^1 \frac{1}{a(y)}dy\right). \tag{4.9}$$

The integrals in Eqs. (4.9) and (4.8) must be obtained by highly accurate numerical integration. For this example, we choose the quantity of interest to be $u(x;\boldsymbol{\xi})$ at $x = 0.35$. We aim to build a 3rd-order Hermite polynomial expansion which includes $N = 816$ basis functions. The relative error is approximated by a level-6 sparse grid method. Fig. 6 shows that accuracy of the re-weighted ℓ_1 (3 iterations) and the iteratively rotated ℓ_1 ($L = 3$ iterations) method are very close in this case. Fig. 6 shows the results of the iterative rotation process applied to the OMP method. In all cases, the incorporation of iterative rotation improves the performance of the other methods. A comparison of $\boldsymbol{c}$ and $\tilde{\boldsymbol{c}}$ are presented in Fig. 7, which shows the improvement of the sparsity in the similar manner as in example function with high compressibility in Sec. 4.2.
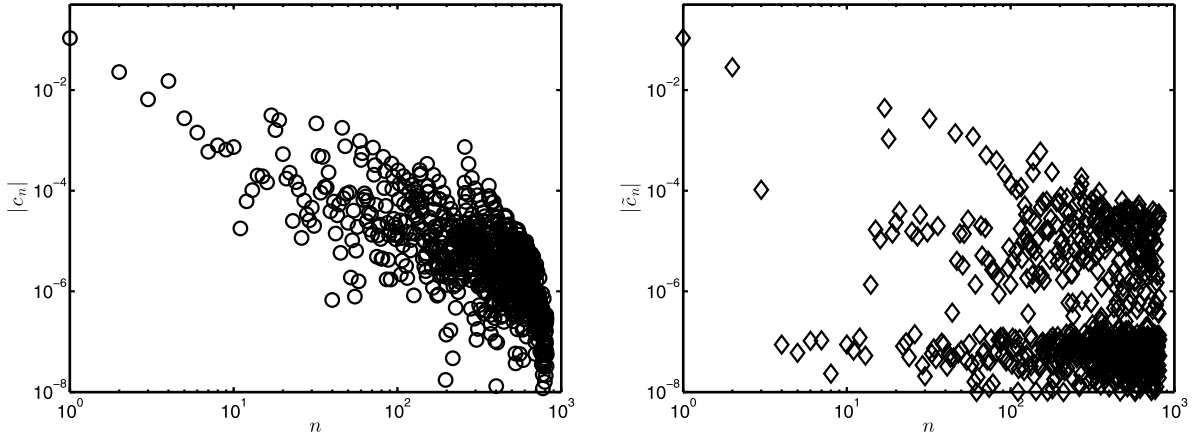
Fig. 7. Magnitude of the gPC coefficients for example elliptic differential equation. (Left) Magnitude of $\mathbf{c}_n$. (Right) Magnitude of $\tilde{\mathbf{c}}_n$ of a randomly chosen replicate computed by re-weighted and iteratively rotated ℓ_1 with $M = 240$ ($M/N \approx 0.3$). These calculations were performed with dimension $d = 15$ and the number of unknowns $N = 816$.

4.4. Example Korteweg–de Vries equation

As an example application of our new method to a more complicated and nonlinear differential equation, we consider the Korteweg–de Vries (KdV) equation with time-dependent additive noise [50]:

$$u_t(x, t; \boldsymbol{\xi}) - 6u(x, t; \boldsymbol{\xi})u_x(x, t; \boldsymbol{\xi}) + u_{xxx}(x, t; \boldsymbol{\xi}) = f(t; \boldsymbol{\xi}), \quad x \in (-\infty, \infty),$$

$$u(x, 0; \boldsymbol{\xi}) = -2\,\text{sech}^2(x). \tag{4.10}$$

Defining

$$W(t; \boldsymbol{\xi}) = \int_0^t f(y; \boldsymbol{\xi})\mathrm{d}y, \tag{4.11}$$

the analytical solution of Eq. (4.10) is

$$u(x, t; \boldsymbol{\xi}) = W(t; \boldsymbol{\xi}) - 2\,\text{sech}^2\left(x - 4t + 6\int_0^t W(z; \boldsymbol{\xi})\mathrm{d}z\right). \tag{4.12}$$

We model $f(t; \boldsymbol{\xi})$ as a Gaussian random field represented by the following KL expansion:

$$f(t; \boldsymbol{\xi}) = \sigma \sum_{i=1}^d \sqrt{\lambda_i}\phi_i(t)\xi_i, \tag{4.13}$$

where σ is a constant and $\{\lambda_i, \phi_i(t)\}_{i=1}^d$ are eigenpairs of the exponential covariance kernel as in Eqs. (4.6) and (4.7), respectively. In this problem, we set $l_c = 0.25$ and $d = 10$ ($\sum_{i=1}^d \lambda_i > 0.96 \sum_{i=1}^\infty \lambda_i$). In this case, the exact one-soliton solution is

$$u(x, t; \boldsymbol{\xi}) = \sigma \sum_{i=1}^d \sqrt{\lambda_i}\xi_i \int_0^t \phi_i(y)\mathrm{d}y - 2\,\text{sech}^2\left(x - 4t + 6\sigma \sum_{i=1}^d \sqrt{\lambda_i}\xi_i \int_0^t \int_0^z \phi_i(y)\mathrm{d}y\mathrm{d}z\right). \tag{4.14}$$

Since an analytical expression for ϕ_i is available, we can compute the integrals in Eq. (4.14) with high accuracy. Denoting

$$A_i = \sqrt{\lambda_i}\int_0^t \phi_i(y)\mathrm{d}y, \quad B_i = \sqrt{\lambda_i}\int_0^t \int_0^z \phi_i(y)\mathrm{d}y\mathrm{d}z, \quad i = 1, 2, \cdots, d, \tag{4.15}$$

the analytical solution is

$$u(x, t; \boldsymbol{\xi}) = \sigma \sum_{i=1}^d A_i \xi_i - 2\,\text{sech}^2\left(x - 4t + 6\sigma \sum_{i=1}^d B_i \xi_i\right). \tag{4.16}$$
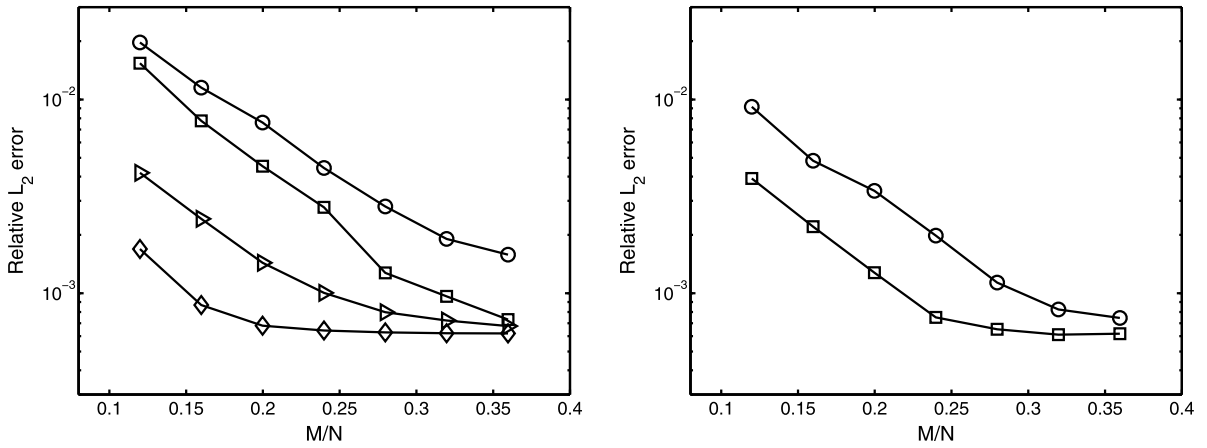
Fig. 8. Results for the example KdV equation. (Left) Comparison to ℓ_1 methods. "∘": standard ℓ_1, "▷": re-weighted ℓ_1, "□": rotated ℓ_1, "◇": re-weighted and iteratively rotated ℓ_1. (Right) Comparison to the OMP method. ∘": standard OMP, "□": rotated OMP. The dimension is $d = 10$ and the number of unknowns is $N = 1001$.

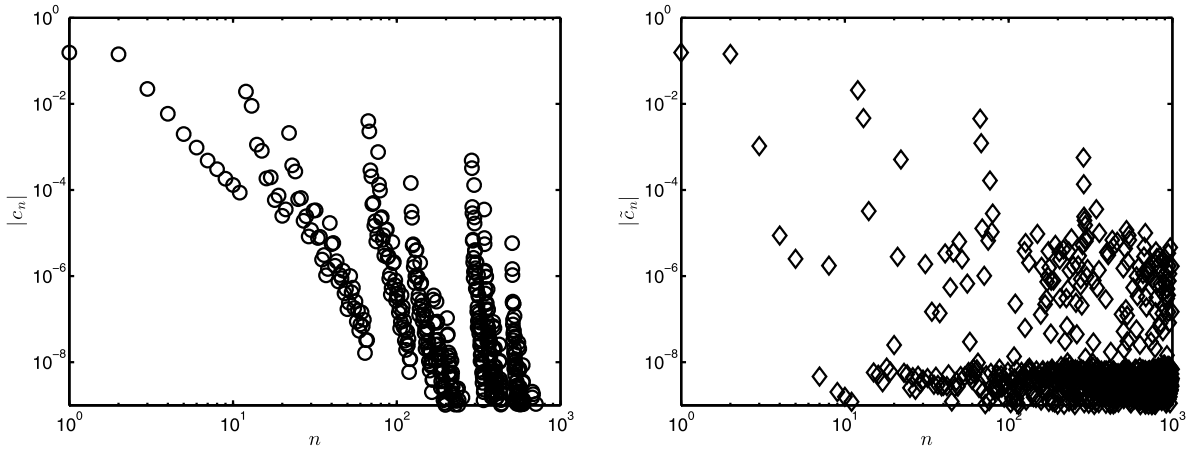**Fig. 9.** Magnitude of the gPC coefficients for example KdV equation. (Left) Magnitude of c_n. (Right) Magnitude of $\tilde{c}_n$ of a randomly chosen replicate computed by re-weighted and iteratively rotated ℓ_1 with $M = 120$ ($M/N \approx 0.12$). These calculations were performed with dimension $d = 10$ and the number of unknowns $N = 1001$.

The quantity of interest is chosen to be $u(x, t; \boldsymbol{\xi})$ at $x = 6, t = 1$ with $\sigma = 0.1$, $P = 4$, and the number of gPC basis functions $N = 1001$. For this example, Fig. 8 shows that the combined iterative rotation and re-weighted ℓ_1 method outperforms all other approaches. However, unlike previous examples the non-rotated re-weighted ℓ_1 method works better than our iteratively rotated unweighted method. This difference likely arises because $\boldsymbol{c}$ is sparser in this case than in others, which makes re-weighted ℓ_1 method more efficient. The pattern of sparsity in this case is different than previous examples, hence the efficiency of identifying a good rotation matrix A is different. A comparison of $\boldsymbol{c}$ and $\tilde{\boldsymbol{c}}$ are presented in Fig. 9, which shows the improvement of the sparsity by the iterative rotation method.

4.5. Example high-dimensional function

The previous examples demonstrate the capability of our new method to solve moderately high-dimensional problems. In the last example, we illustrate its potential for dealing with higher-dimensional problems. Specially, we select a function similar to the first example (Sec. 4.1) but with much higher dimensionality:

$$u(\boldsymbol{\xi}) = \sum_{i=1}^{d} \xi_i + 0.25 \left(\sum_{i=1}^{d} \xi_i / \sqrt{i} \right)^2, \quad d = 100. \tag{4.17}$$

The total number of basis functions for this example is $N = 5151$. The relative error is computed with a level-3 sparse grid method, hence the numerical integrals are exact. The results are presented in Fig. 10. As before, our iterative rotation approach out-performs the existing ℓ_1 and OMP methods. A comparison of $\boldsymbol{c}$ and $\tilde{\boldsymbol{c}}$ is presented in Fig. 11 and it shows the enhancement of the sparsity by the iterative rotation method.

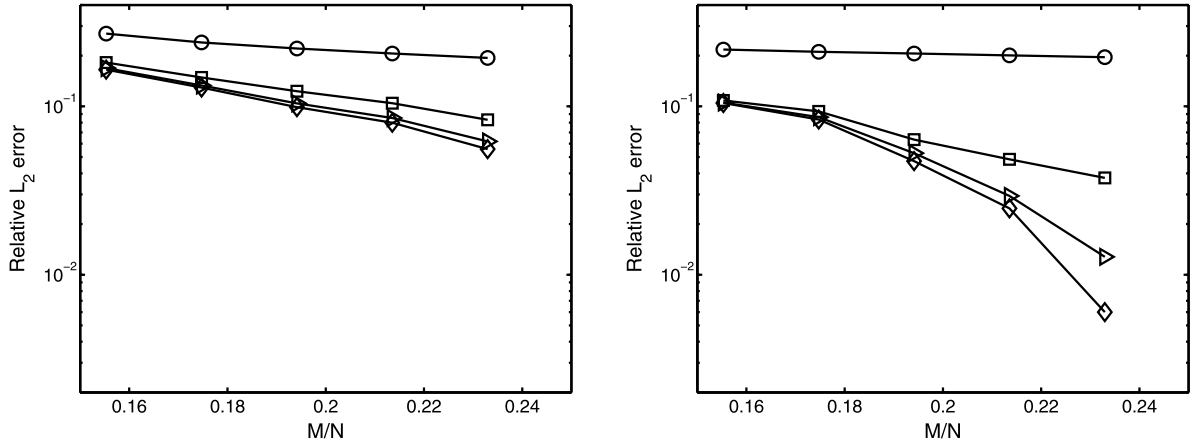**Fig. 10.** Results for the example high-dimensional function. (Left) Comparison with ℓ_1 methods. "○": standard ℓ_1, "□": rotated ℓ_1 with 1 iterations, "▷": rotated ℓ_1 with 2 iterations; "◇": rotated ℓ_1 with 3 iterations. (Right) Comparison with OMP methods. "□": rotated OMP with 1 iterations, "▷": rotated OMP with 2 iterations; "◇": rotated OMP with 3 iterations. These calculations were performed with $d = 100$ and the number of unknowns $N = 5151$.
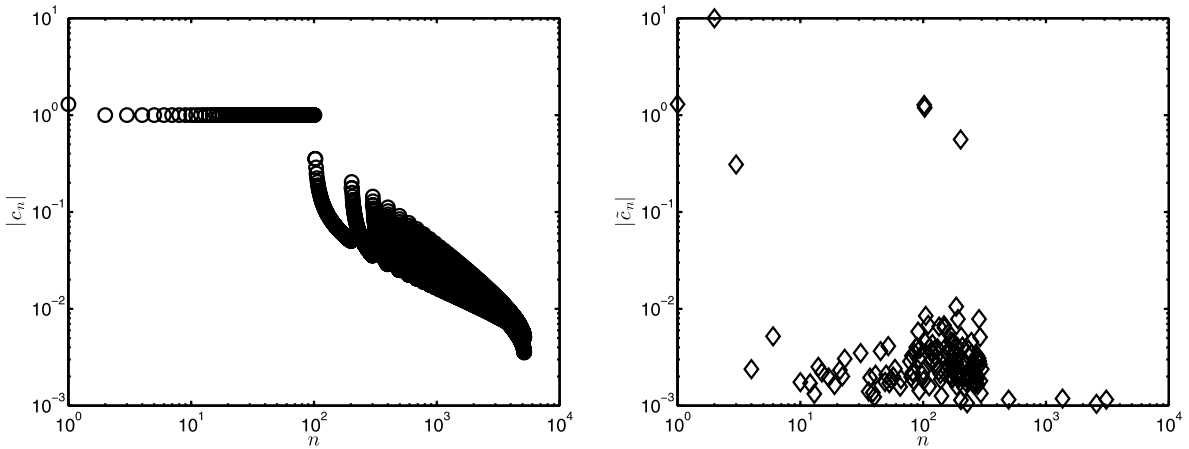

Fig. 11. Magnitude of the gPC coefficients for example high-dimensional function. (Left) Magnitude of $\mathbf{c}_n$. (Right) Magnitude of $\tilde{\mathbf{c}}_n$ of a randomly chosen replicate computed by 3 iterated OMP method with $M = 1200$ ($M/N \approx 0.23$). These calculations were performed with dimension $d = 100$ and the number of unknowns $N = 5151$.

For general high-dimensional problems, simply truncating the gPC expansion up to a certain order is not efficient because the number of basis function will be very large. For example, in this test, $P = 2$ requires 5151 basis functions. Under such conditions even a small $M/N = 0.2$ needs 1030 samples, which can be difficult in practical problems when the computational model is very costly. Hence, a good approach for high-dimensional problems is to integrate our iterative rotation method with an adaptive method to reduce N; e.g., adaptive basis selection [22] or an ANOVA method [47].

4.6. Accuracy of computing the expansion coefficients $\mathbf{c}$

In many applications, gPC expansions are also used to study the sensitivity of the quantity of interest to the input random variables $\boldsymbol{\xi}$. In order to perform this analysis, we need to transform the $u_g(\boldsymbol{\eta}) = \sum_{n=1}^{N} \tilde{c}_n \psi_n(\boldsymbol{\eta})$ back to the original variables $u_g(\boldsymbol{\xi}) = \sum_{n=1}^{N} c_n \psi_n(\boldsymbol{\xi})$. This transformation can be accomplished through inversion of A in $\boldsymbol{\eta} = A\boldsymbol{\xi}$. In Figs. 12 and 13, we present the coefficients of $u_g(\boldsymbol{\xi})$ in Examples 4.1 and 4.4, respectively. In both figures we randomly choose one test from the 100 replicates. In Fig. 12, we select a result with $M = 180$ ($M/N \approx 0.4$). Using the standard ℓ_1 method (left) gives very inaccurate results for $\mathbf{c}$. However, Fig. 12 (right) shows that the iterative rotation method with $L = 9$ iterations gives much more accurate results for $\mathbf{c}$. We observe the same behavior in Fig. 13, where we chose a test with $M = 120$ ($M/N \approx 0.12$) for the example KdV equation (Sec. 4.4). In order to make this figure legible, we only present those c_i with absolute value larger than 10^{-5}. This example demonstrates that coefficients c_n with magnitude larger than 10^{-3} are computed accurately by the combined iterative rotation and re-weighted ℓ_1 method while the standard ℓ_1 obtained significantly less accurate c_n. This difference is more distinct for $|c_n| \in [10^{-4}, 10^{-3}]$; in this range our new method compute c_n much more accurate than the standard ℓ_1 method. In the lower right corner of the left plot, the standard ℓ_1 method yields many c_n which should not appear in that area. As a comparison, we do not see such c_n calculated with the new method.
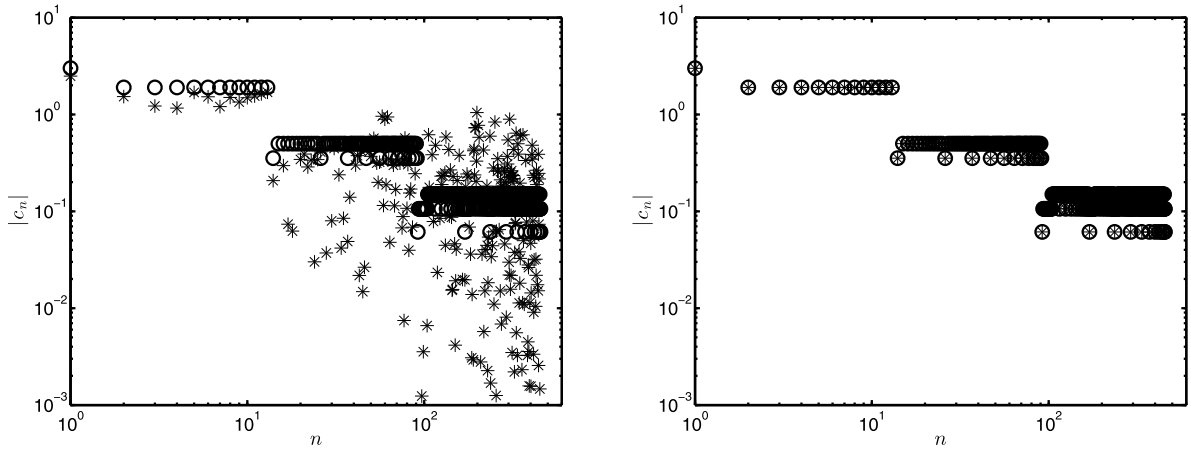
Fig. 12. Comparison of gPC coefficients for the example function with equally important random variables (Sec. 4.1). (Left) Coefficients calculated by the standard ℓ_1 method. "○": exact $|c_i|$; "∗": $|c_i|$ by standard ℓ_1 method. (Right) Coefficients calculated by our new iteratively rotated ℓ_1 method with $L = 9$ iterations.
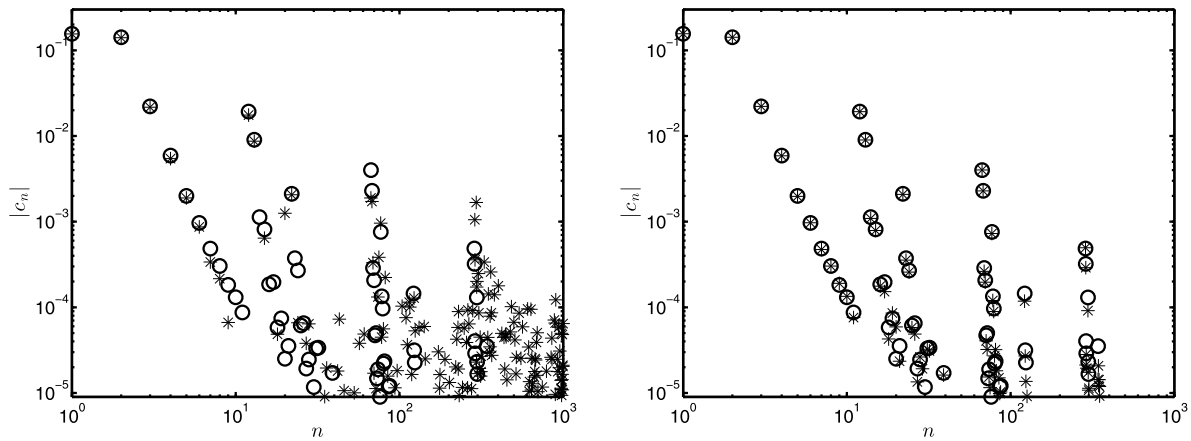

Fig. 13. Comparison of gPC coefficients for solutions of the KdV equation (Sec. 4.4). (Left) Coefficients calculated with the standard ℓ_1 method. "○": "exact" $|c_i|$; "∗": $|c_i|$ by standard ℓ_1 method. (Right) Coefficients calculated by our new iteratively rotated re-weighted ℓ_1 method (right). Only $|c_i| > 10^{-5}$ are presented.

5. Conclusions

In this paper, we extend our previous work [17] and have introduced a compressive sensing-based gPC method to increase the sparsity and accuracy of Hermite polynomial expansion with iterative rotations. Similar to the active subspace method [28,17], the rotation is decided by seeking the directions of maximum variation for the quantity of interest. Our current numerical examples are intended to demonstrate the ability of the method to increase the sparsity and accuracy of the gPC expansion; therefore, the quantity of interest only relies on the random variables. It is also possible to include the physical variables in the basis functions, i.e., $u(\boldsymbol{x}; \boldsymbol{\xi}) = \sum_n c_n \psi_n(\boldsymbol{x}; \boldsymbol{\xi})$ (e.g., [16]), our future work will explore how this new method may help to increase the sparsity in such cases.

We have demonstrated the method for ℓ_1 minimization and OMP methods but it can also be integrated with other compressive sensing methods. In particular, future work will investigate the integration of our new methods with advanced sampling strategies (e.g., [20]), adaptive basis selection method (e.g., [22]), Bayesian compressive sensing method (e.g., [16]), etc. These advanced strategies are particularly important for high-dimensional problems.

With this method, we will also be able to construct an accurate surrogate model of the quantity of interest with limited data. Surrogate models are specifically useful for the problems where the experiments or simulations are very costly. This surrogate model can be used to study the sensitivity of the parameters and is very useful in inverse problems based on Bayesian framework. Our new method requires fewer output data to construct such surrogate models, which can be a great savings of experimental or computational resources.

Finally, we highlight three additional areas of future work for improving the new method. First, it is currently only suitable for Hermite polynomial expansions. Second, the new method requires a formal numerical analysis to assess convergence behavior and determine specific terminating criteria. Finally, there are likely more optimal iterative rotation strategies

that can be applied to Hermite polynomial or other expansions. One possible direction of work is to design a suitable objective function and consider this problem from an optimization point of view. All of these questions will be addressed in our future work.

Acknowledgements

Appendix A

Here we provide the details of computing the elements $(K_{ij})_{kl}$ in Eq. (3.6). Notice that for univariate normalized Hermite polynomials,

$$\psi_n(\xi)' = \sqrt{n}\psi_{n-1}(\xi), \quad n \in \mathbb{N} \cup \{0\}, \tag{A.1}$$

where we set $\psi_{-1}(\xi) = 0$ for simplicity. Therefore, we have

$$\mathbb{E}\left\{\psi_i(\xi)'\psi_j(\xi)\right\} = \int_{\mathbb{R}} \psi_i(\xi)'\psi_j(\xi)\rho(\xi)\mathrm{d}\xi = \sqrt{i}\delta_{i-1\,j}, \tag{A.2}$$

where $\rho(\xi)$ is the PDF of a standard Gaussian random variable. For a multi-index $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \cdots, \alpha_d), \alpha_i \in \mathbb{N} \cup \{0\}$, and basis function $\psi_{\boldsymbol{\alpha}}(\boldsymbol{\xi}) = \psi_{\alpha_1}(\xi_1)\psi_{\alpha_2}(\xi_2)\cdots\psi_{\alpha_d}(\xi_d)$,

$$\frac{\partial}{\partial \xi_i}\psi_{\boldsymbol{\alpha}}(\boldsymbol{\xi}) = \psi_{\alpha_i}(\xi_i)'\prod_{\substack{m=1 \\ m \neq i}}^{d} \psi_{\alpha_m}(\xi_m). \tag{A.3}$$

Hence, given two different multi-indices $\boldsymbol{\alpha}_k = ((\alpha_k)_1, (\alpha_k)_2, \cdots, (\alpha_k)_d)$ and $\boldsymbol{\alpha}_l = ((\alpha_l)_1, (\alpha_l)_2, \cdots, (\alpha_l)_d)$, the corresponding entry of matrix K_{ij} is

$$\begin{aligned}
(K_{ij})_{kl} &= \mathbb{E}\left\{\frac{\partial \psi_{\boldsymbol{\alpha}_k}(\boldsymbol{\xi})}{\partial \xi_i} \cdot \frac{\partial \psi_{\boldsymbol{\alpha}_l}(\boldsymbol{\xi})}{\partial \xi_j}\right\} \\
&= \mathbb{E}\left\{\left(\psi_{(\alpha_k)_i}(\xi_i)'\prod_{\substack{m=1 \\ m \neq i}}^{d} \psi_{(\alpha_k)_m}(\xi_m)\right) \cdot \left(\psi_{(\alpha_l)_j}(\xi_j)'\prod_{\substack{m=1 \\ m \neq j}}^{d} \psi_{(\alpha_l)_m}(\xi_m)\right)\right\} \\
&= \sqrt{(\alpha_k)_i(\alpha_l)_j}\,\delta_{(\alpha_k)_i-1(\alpha_l)_i}\delta_{(\alpha_k)_j(\alpha_l)_j-1} \cdot \prod_{\substack{m=1 \\ m \neq i, m \neq j}} \delta_{(\alpha_k)_m(\alpha_l)_m}.
\end{aligned} \tag{A.4}$$

References

[1] R.G. Ghanem, P.D. Spanos, Stochastic Finite Elements: A Spectral Approach, Springer-Verlag, New York, 1991.
[2] D. Xiu, G.E. Karniadakis, The Wiener–Askey polynomial chaos for stochastic differential equations, SIAM J. Sci. Comput. 24 (2) (2002) 619–644.
[3] R.H. Cameron, W.T. Martin, The orthogonal development of non-linear functionals in series of Fourier–Hermite functionals, Ann. Math. (1947) 385–392.
[4] H. Ogura, Orthogonal functionals of the Poisson process, IEEE Trans. Inf. Theory 18 (4) (1972) 473–481.
[5] M.A. Tatang, W. Pan, R.G. Prinn, G.J. McRae, An efficient method for parametric uncertainty analysis of numerical geophysical models, J. Geophys. Res., Atmos. 102 (D18) (1997) 21925–21932.
[6] D. Xiu, J.S. Hesthaven, High-order collocation methods for differential equations with random inputs, SIAM J. Sci. Comput. 27 (3) (2005) 1118–1139.
[7] J. Foo, X. Wan, G.E. Karniadakis, The multi-element probabilistic collocation method (ME-PCM): error analysis and applications, J. Comput. Phys. 227 (22) (2008) 9572–9595.
[8] I. Babuška, F. Nobile, R. Tempone, A stochastic collocation method for elliptic partial differential equations with random input data, SIAM Rev. 52 (2) (2010) 317–355.
[9] E.J. Candès, J.K. Romberg, T. Tao, Stable signal recovery from incomplete and inaccurate measurements, Commun. Pure Appl. Math. 59 (8) (2006) 1207–1223.
[10] D.L. Donoho, M. Elad, V.N. Temlyakov, Stable recovery of sparse overcomplete representations in the presence of noise, IEEE Trans. Inf. Theory 52 (1) (2006) 6–18.
[11] E.J. Candès, The restricted isometry property and its implications for compressed sensing, C. R. Math. Acad. Sci. Paris 346 (9–10) (2008) 589–592.
[12] A.M. Bruckstein, D.L. Donoho, M. Elad, From sparse solutions of systems of equations to sparse modeling of signals and images, SIAM Rev. 51 (1) (2009) 34–81.
[13] A. Doostan, H. Owhadi, A non-adapted sparse approximation of PDEs with stochastic inputs, J. Comput. Phys. 230 (8) (2011) 3015–3034.
[14] L. Yan, L. Guo, D. Xiu, Stochastic collocation algorithms using ℓ_1-minimization, Int. J. Uncertain. Quantificat. 2 (3) (2012) 279–293.

[15] X. Yang, G.E. Karniadakis, Reweighted ℓ_1 minimization method for stochastic elliptic differential equations, J. Comput. Phys. 248 (1) (2013) 87–108.
[16] G. Karagiannis, B.A. Konomi, G. Lin, A Bayesian mixed shrinkage prior procedure for spatial–stochastic basis selection and evaluation of GPC expansions: applications to elliptic SPDEs, J. Comput. Phys. 284 (2015) 528–546.
[17] H. Lei, X. Yang, B. Zheng, G. Lin, N. Baker, Constructing surrogate models of complex systems with enhanced sparsity: quantifying the influence of conformational uncertainty in biomolecular solvation, Multiscale Model. Simul. 13 (4) (2015) 1327–1353.
[18] J. Peng, J. Hampton, A. Doostan, A weighted ℓ_1-minimization approach for sparse polynomial chaos expansions, J. Comput. Phys. 267 (2014) 92–111.
[19] Z. Xu, T. Zhou, On sparse interpolation and the design of deterministic interpolation points, SIAM J. Sci. Comput. 36 (4) (2014) A1752–A1769.
[20] J. Hampton, A. Doostan, Compressive sampling of polynomial chaos expansions: convergence analysis and sampling strategies, J. Comput. Phys. 280 (2015) 363–386.
[21] G. Tang, G. Iaccarino, Subsampled Gauss quadrature nodes for estimating polynomial chaos expansions, SIAM J. Uncertain. Quantificat. 2 (1) (2014) 423–443.
[22] J.D. Jakeman, M.S. Eldred, K. Sargsyan, Enhancing ℓ_1-minimization estimates of polynomial chaos expansions using basis selection, J. Comput. Phys. 289 (2015) 18–34.
[23] 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 Quantification 4 (1) (2014) 63–93.
[24] J. Peng, J. Hampton, A. Doostan, On polynomial chaos expansion via gradient-enhanced ℓ_1-minimization, arXiv:1506.00343.
[25] E.J. Candès, M.B. Wakin, S.P. Boyd, Enhancing sparsity by reweighted l_1 minimization, J. Fourier Anal. Appl. 14 (5–6) (2008) 877–905.
[26] H. Rauhut, R. Ward, Interpolation via weighted ℓ_1 minimization, Appl. Comput. Harmon. Anal. (2015).
[27] H. Rauhut, R. Ward, Sparse Legendre expansions via ℓ_1-minimization, J. Approx. Theory 164 (5) (2012) 517–533.
[28] P.G. Constantine, E. Dow, Q. Wang, Active subspace methods in theory and practice: applications to kriging surfaces, SIAM J. Sci. Comput. 36 (4) (2014) A1500–A1524.
[29] S.S. Chen, D.L. Donoho, M.A. Saunders, Atomic decomposition by basis pursuit, SIAM J. Sci. Comput. 20 (1) (1998) 33–61.
[30] S. Boyd, L. Vandenberghe, Convex Optimization, Cambridge University Press, 2004.
[31] D.L. Donoho, Compressed sensing, IEEE Trans. Inf. Theory 52 (4) (2006) 1289–1306.
[32] E.J. Candès, T. Tao, Decoding by linear programming, IEEE Trans. Inf. Theory 51 (12) (2005) 4203–4215.
[33] D. Needell, Noisy signal recovery via iterative reweighted ℓ_1 minimization, in: Proc. Asilomar Conf. on Signal Systems and Computers, Pacific Grove, CA, 2009.
[34] J.A. Tropp, Greed is good: algorithmic results for sparse approximation, IEEE Trans. Inf. Theory 50 (10) (2004) 2231–2242.
[35] J.A. Tropp, A.C. Gilbert, Signal recovery from random measurements via orthogonal matching pursuit, IEEE Trans. Inf. Theory 53 (12) (2007) 4655–4666.
[36] J.W. Hardin, J. Hilbe, Generalized Linear Models and Extensions, Stata Press, 2007.
[37] Y. Xia, A constructive approach to the estimation of dimension reduction directions, Ann. Stat. (2007) 2654–2690.
[38] O. Roderick, M. Anitescu, P. Fischer, Polynomial regression approaches using derivative information for uncertainty quantification, Nucl. Sci. Eng. 164 (2) (2010) 122–139.
[39] Y. Li, M. Anitescu, O. Roderick, F. Hickernell, Orthogonal bases for polynomial regression with derivative information in uncertainty quantification, Int. J. Uncertainty Quantification 1 (4) (2011) 297–320.
[40] B. Lockwood, D. Mavriplis, Gradient-based methods for uncertainty quantification in hypersonic flows, Comput. Fluids 85 (2013) 27–38.
[41] S. Smolyak, Quadrature and interpolation formulas for tensor products of certain classes of functions, Sov. Math. Dokl. 4 (1963) 240–243.
[42] E. van den Berg, M.P. Friedlander, Probing the Pareto frontier for basis pursuit solutions, SIAM J. Sci. Comput. 31 (2) (2008) 890–912.
[43] E. van den Berg, M.P. Friedlander, SPGL1: a solver for large-scale sparse reconstruction, http://www.cs.ubc.ca/labs/scl/spgl1, June 2007.
[44] D. Donoho, I. Drori, V. Stodden, Y. Tsaig, Sparselab: seeking sparse solutions to linear systems of equations, http://www-stat.stanford.edu/~sparselab/.
[45] X. Ma, N. Zabaras, An adaptive hierarchical sparse grid collocation algorithm for the solution of stochastic differential equations, J. Comput. Phys. 228 (8) (2009) 3084–3113.
[46] X. Ma, N. Zabaras, An adaptive high-dimensional stochastic model representation technique for the solution of stochastic partial differential equations, J. Comput. Phys. 229 (10) (2010) 3884–3915.
[47] X. Yang, M. Choi, G. Lin, G.E. Karniadakis, Adaptive ANOVA decomposition of stochastic incompressible and compressible flows, J. Comput. Phys. 231 (4) (2012) 1587–1614.
[48] Z. Zhang, X. Yang, G. Marucci, P. Maffezzoni, I.A. Elfadel, G. Karniadakis, L. Daniel, Stochastic testing simulator for integrated circuits and MEMS: hierarchical and sparse techniques, in: IEEE Proceedings of the Custom Integrated Circuits Conference, CICC 2014, 2014, pp. 1–8.
[49] M. Jardak, C.-H. Su, G.E. Karniadakis, Spectral polynomial chaos solutions of the stochastic advection equation, J. Sci. Comput. 17 (1–4) (2002) 319–338.
[50] G. Lin, L. Grinberg, G.E. Karniadakis, Numerical studies of the stochastic Korteweg–de Vries equation, J. Comput. Phys. 213 (2) (2006) 676–703.