A GENERAL FRAMEWORK OF ENHANCING SPARSITY OF GENERALIZED POLYNOMIAL CHAOS EXPANSIONS

XIU YANG†, XIAOLIANG WAN, AND LIN LIN

Abstract. Compressive sensing has become a powerful addition to uncertainty quantification when only limited data is available. In this paper we provide a general framework to enhance the sparsity of the representation of uncertainty in the form of generalized polynomial chaos expansion. We use alternating direction method to identify new sets of random variables through iterative rotations such that the new representation of the uncertainty is sparser. Consequently, we increases both the efficiency and accuracy of the compressive sensing-based uncertainty quantification method. We demonstrate that the previously developed iterative method to enhance the sparsity of Hermite polynomial expansion is a special case of this general framework. Moreover, we use Legendre and Chebyshev polynomials expansions to demonstrate the effectiveness of this method with applications in solving stochastic partial differential equations and high-dimensional (O(100)) problems.

Keywords: compressive sensing, uncertainty quantification, iterative rotation, alternating direction.

1. Introduction

Surrogate model based 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 (QoI). In this paper, we focus on parametric uncertainty, i.e., some of the parameters in the system are random variables. We assume that these random variables are defined on a probability space \((\Omega, \mathcal{F}, P)\), where \(\Omega\) is the event space and \(P\) is a probability measure on the \(\sigma\)-field \(\mathcal{F}\). We consider a system depending on a \(d\)-dimensional random vector \(\mathbf{\xi}(\omega) = (\xi_1(\omega), \xi_2(\omega), \cdots, \xi_d(\omega))^T\), where \(\omega\) is an event in \(\Omega\). For simplicity, we denote \(\xi_i(\omega)\) as \(\xi_i\). A very useful surrogate model of the QoI \(u(\mathbf{\xi})\) is the generalized polynomial chaos (gPC) expansion \[16, 42\]:

\[
u(\mathbf{\xi}) = \sum_{n=1}^{N} c_n \psi_n(\mathbf{\xi}) + \varepsilon(\mathbf{\xi}),
\]

where \(\varepsilon\) is the truncation error, \(N\) is a positive integer, \(c_n\) are coefficients, \(\psi_n\) are multivariate polynomials which are orthonormal with respect to the distribution of \(\mathbf{\xi}\):

\[
\int_{\mathbb{R}^d} \psi_i(x) \psi_j(x) \rho_{\mathbf{\xi}}(x) dx = \delta_{ij},
\]

where \(\rho_{\mathbf{\xi}}(x)\) is the probability density function (PDF) of \(\mathbf{\xi}\) and \(\delta_{ij}\) is the Kronecker delta function. This approximation converges in the \(L_2\) sense as \(N\) increases if \(u\) is in the Hilbert space associated with the measure of \(\mathbf{\xi}\) (i.e., the weight of the inner product is the PDF of \(\mathbf{\xi}\) \[42, 5, 27\]. Both intrusive method (e.g., stochastic Galerkin) non-intrusive method (e.g., probabilistic collocation method) (PCM)) are developed \[16, 42, 36, 41, 14, 3\] to compute the gPC coefficients \(c = (c_1, c_2, \cdots, c_N)^T\). Specifically, the non-intrusive method is more suitable...
for the study of complex system as it does not need to modify the computational model of the system while the intrusive method requires rewriting the existing model. The non-intrusive method uses the samples of input \( \{ \xi_q \}_{q=1}^M \) and corresponding output of the computational model \( u^d = u(\xi^q) \) to compute the \( c \).

Note that in many practical problems, it is very costly to obtain \( u^d \) 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:

\[
\Psi c = u - \varepsilon, \tag{3}
\]

where \( u = (u^1, u^2, \cdots, u^M)^T \) is the vector of output samples, \( c = (c_1, c_2, \cdots, c_N)^T \) is the vector of gPC coefficients, \( \Psi \) is an \( M \times N \) matrix with \( \Psi_{ij} = \psi_j(\xi^i) \) and \( \varepsilon = (\varepsilon^1, \varepsilon^2, \cdots, \varepsilon^M)^T \) is a vector of error samples with \( \varepsilon^q = \varepsilon(\xi^q) \). The compressive sensing method is effective at solving this type of under-determined problem when \( c \) is sparse \( [14, 42, 3, 4] \) and recent studies have applied this approach to uncertainty quantification (UQ) problems \( [13, 44, 46, 23, 43, 33, 29, 22] \).

Several useful approaches have been developed to improve the efficiency of solving Eq. (3) in UQ applications. For example, weighted/re-weighted \( \ell_1 \) minimization assigns a weight to each \( c_n \) and solves a weighted \( \ell_1 \) minimization problem to enhance the sparsity \( [9, 10, 23, 31] \); better sampling strategies can improve the property of \( \Psi \) \( [30, 17, 2] \); adaptive basis selection reduces the number of unknown \( [19] \). Recently, we proposed an approach to enhance the sparsity of \( c \) through the rotation of the random vector \( \xi \) \( [23, 47] \).

In this work we provide a general framework to enhance the sparsity of the representation of uncertainty in the form of generalized polynomial chaos expansion. We use alternating direction method to iteratively identify a rotation \( g : \mathbb{R}^d \mapsto \mathbb{R}^d \) which maps \( \xi \) to a new set of random variables \( \eta = (\eta_1, \eta_2, \cdots, \eta_d)^T \) such that the gPC expansion of \( u \) with respect to \( \eta \) is sparser. In other words,

\[
u(\xi) \approx \sum_{n=1}^{N} c_n \psi_n(\xi) = \sum_{n=1}^{N} \tilde{c}_n \psi_n(\eta(\xi)) \approx u(\eta(\xi)), \tag{4}\]

and we intend to make \( \tilde{c} = (\tilde{c}_1, \tilde{c}_2, \cdots, \tilde{c}_N)^T \) sparser than \( c \). By increasing the sparsity, we can improve both the efficiency and accuracy of the compressive sensing-based uncertainty quantification method in that we can use fewer samples of \( u^d \) to obtain a more accurate gPC expansion. We demonstrate that the previously developed iterative method to enhance the sparsity of Hermite polynomial expansion \( [17] \) is a special case of this general framework. We will use Legendre polynomial expansion and Chebyshev polynomial expansion to demonstrate the effectiveness of our proposed method.

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

2.1. Generalized polynomial chaos expansions. In this paper we study systems relying on \( d \)-dimensional i.i.d. random variables \( \xi \), hence, the gPC basis functions are constructed by tensor products of univariate orthonormal polynomials. For a multi-index \( \alpha = (\alpha_1, \alpha_2, \cdots, \alpha_d), \alpha_i \in \mathbb{N} \cup \{0\} \), we set

\[
\psi_\alpha(\xi) = \psi_{\alpha_1}(\xi_1) \psi_{\alpha_2}(\xi_2) \cdots \psi_{\alpha_d}(\xi_d). \tag{5}\]

For two different multi-indices \( \alpha_i = ((\alpha_{i_1}), (\alpha_{i_2}), \cdots, (\alpha_{i_j})) \) and \( \alpha_j = ((\alpha_{j_1}), (\alpha_{j_2}), \cdots, (\alpha_{j_d})) \), we have the property

\[
\int_{\mathbb{R}^d} \psi_{\alpha_i}(x) \psi_{\alpha_j}(x) \rho_\xi(x) dx = \delta_{\alpha_i, \alpha_j} = \delta_{\alpha_{i_1}, \alpha_{j_1}} \delta_{\alpha_{i_2}, \alpha_{j_2}} \cdots \delta_{\alpha_{i_d}, \alpha_{j_d}}, \tag{6}\]

where

\[
\rho_\xi(x) = \rho_{\xi_1}(x_1) \rho_{\xi_2}(x_2) \cdots \rho_{\xi_d}(x_d). \tag{7}\]

For simplicity, we denote \( \psi_{\alpha_i}(\xi) \) as \( \psi_i(\xi) \).
2.2. Compressive sensing. We first introduce the concept of sparsity as it is critical in the error estimates for solving the under-determined system Eq. (3) with the compressive sensing method. The $\ell_0$ “norm” of vector $x = (x_1, x_2, \cdots, x_N)$ is defined as the number of its non-zero entries \[\|x\|_0 \overset{\text{def}}{=} \# \{i : x_i \neq 0\} \] (8) and $\ell_1$ norm is defined as the sum of the absolute value of its entries: \[\|x\|_1 \overset{\text{def}}{=} \sum_{n=1}^N |x_n|. \] (9) $x$ 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 $\ell_1$ 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 \[6\].

Under some conditions, the sparse vector $c$ in Eq. (3) can be approximated by solving the following $\ell_1$ minimization problem:

\[
(P_{1,\epsilon}): \quad \arg \min_{\hat{c}} \|\hat{c}\|_1, \text{ subject to } \|\Psi \hat{c} - u\|_2 \leq \epsilon, \tag{10}
\]

where $\epsilon = \|\varepsilon\|_2$. The error bound for solving Eq. (3) with $\ell_1$ minimization requires definition of the restricted isometry property (RIP) constant \[8\]. For each integer $s = 1, 2, \cdots$, the isometry constant $\delta_s$ of a matrix $\Phi$ is defined as the smallest number such that

\[
(1 - \delta_s)\|x\|_2^2 \leq \|\Phi x\|_2^2 \leq (1 + \delta_s)\|x\|_2^2 \tag{11}
\]

holds for all $s$-sparse vectors $x$. With some restrictions, Candès et al. showed $x$ can be stably reconstructed \[6\]. Assume that the matrix $\Psi$ satisfies $\delta_{2s} < \sqrt{2} - 1$, and $\|\varepsilon\|_2 \leq \epsilon$, then solution $\hat{c}$ to $(P_{1,\epsilon})$ obeys

\[
\|c - \hat{c}\|_2 \leq C_1 \epsilon + C_2 \frac{\|c - c_s\|_1}{\sqrt{s}}, \tag{12}
\]

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. (12), respectively. We will use $\|c - c_s\|_1/\sqrt{s}$ to examine the sparsity in the our numerical examples.

The re-weighted $\ell_1$ minimization approach is an improvement of the $\ell_1$ minimization method, which enhances the accuracy of estimating $c$ \[11\]. This approach solves the following optimization problem:

\[
(P_{1W}^\epsilon): \quad \arg \min_{\hat{c}} \|W \hat{c}\|_1, \text{ subject to } \|\Psi \hat{c} - u\|_2 \leq \epsilon, \tag{13}
\]

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 $(P_{1W}^\epsilon)$ by setting $W = I$. The elements $w_i$ of the diagonal matrix can be estimated iteratively \[11\] [46]. More precisely, for each iteration $l$, $(P_{1W}^\epsilon)$ 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 Candès et al. \[9\], the authors suggest two to three iterations of this procedure. Subsequent analytical work \[24\] provides an error bound for each iteration as well as the limit of computing $\hat{c}$ with re-weighted $\ell_1$ minimization. The form is similar to Eq. (12) with different constants.

In practice, the error term $\epsilon$ is not known a priori, hence in the present work we use cross-validation to estimate it. One such algorithm is \[13\] summarized in Algorithm 1. We note
that some techniques may be applied to avoid the cross-validation step and we refer interested readers to [1].

2.3. Compressive sensing-based gPC methods. Given $M$ samples of $\xi$, the quantity of interest $u$ is approximated by a gPC expansion as in Eq. (1):

$$u(\xi^q) = \sum_{n=1}^{N} c_n \psi_n(\xi^q) + \epsilon(\xi^q), \quad q = 1, 2, \cdots, M,$$

which can be rewritten as Eq. (3). A typical approach to compressive sensing based-gPC is summarized in Algorithm 2:

**Algorithm 2** Compressive sensing-based gPC

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

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

where $h = 0$ or $1$, $u = (u^1, u^2, \cdots, u^M)^T$, and $\epsilon$ is obtained by cross-validation. If the re-weighted $\ell_1$ method is employed, solve ($P_{h,\epsilon}^{\ell_1}$) instead.
5. Set $c = \hat{c}$ and construct gPC expansion as $u(\xi) \approx \sum_{n=1}^{N} c_n \psi_n(\xi)$.

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 [4]:

$$\mu(\Psi) = \max_{1 \leq j, k \leq N, j \neq k} \frac{\|\Psi_j^T \Psi_k\|_1}{\|\Psi_j\|_2 \cdot \|\Psi_k\|_2},$$

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

3. Iterative rotations

3.1. Basic idea. We aim to find a linear mapping $g : \mathbb{R}^d \rightarrow \mathbb{R}^d$ such that we have a new set of random variables:

$$\eta = g(\xi) = A \xi, \quad \eta = (\eta_1, \eta_2, \cdots, \eta_d)^T,$$
where $A$ is an orthogonal matrix satisfying $AA^T = I$ and the PDF of $\eta$ is denoted as $\rho_\eta$. As such, the new polynomial expansion for $u$ is

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

(17)

where

$$u(\xi) = v(\eta(\xi)) = v(\eta) \approx v_g(\eta),$$

$u_g(\xi)$ is understood as a polynomial $u_g(x)$ evaluated at the random variables $\xi$ and the same for $v_g$. Ideally, $\tilde{c}$ is sparser than $c$. In the previous work \cite{47}, we assume that $\xi \sim \mathcal{N}(0, I)$, hence, $\eta \sim \mathcal{N}(0, I)$. For general cases, in which $\{\xi_i\}_{i=1}^d$ are not i.i.d. Gaussian, $\{\eta_i\}_{i=1}^d$ are not necessarily independent. Moreover, $\{\psi_i\}_{i=1}^N$ are not necessarily orthogonal to each other with respect to $\rho_\eta$. Therefore $v_g(\eta)$ may not be a standard gPC expansion of $v(\eta)$. It is simply a polynomial equivalent to $u_g(\xi)$ with potentially sparser coefficients.

Following the idea of ”active subspace” \cite{32, 10}, we use the information of gradient of $u$ to identify the rotation matrix $A$. More specifically, let

$$W = [\nabla u(\xi^1), \nabla u(\xi^2), \ldots, \nabla u(\xi^M)],$$

(18)

where $M$ is the number of available input samples, $\nabla u(x) = (\partial u/\partial x_1, \partial u/\partial x_2, \ldots, \partial u/\partial x_d)^T$ is a column vector and $\nabla u(\xi^j) = \nabla u(x)|_{x = \xi^j}$. The singular value decomposition (SVD) or principle component analysis (PCA) of $W$ yields

$$W = U_W \Sigma_W V_W^T,$$

(19)

where $U_W$ is an orthonormal matrix and $\Sigma_W$ is a diagonal matrix consisting of singular values $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_d \geq 0$. We set the rotation matrix as $A = U_W^T$. As such, the rotation projects $\xi$ to the directions of principle components of $\nabla u$. We note that this is consistent with our previous work using the idea of active subspace \cite{47}, in which we computed the eigen-decomposition of the “stiffness matrix”:

$$G \defeq \mathbb{E} \{ \nabla u(\xi) \otimes \nabla u(\xi) \} = U \Lambda U^T, \quad UU^T = I,$$

(20)

where $G$ is symmetric, $U$ is an orthogonal matrix consists of eigenvectors, $\Lambda = \text{diag}(\lambda_1, \lambda_2, \cdots, \lambda_d)$ with $\lambda_1 \geq \lambda_2 \geq \cdots \geq 0$ and we set $A = U^T$. As pointed out in \cite{10}, $G$ is the variance matrix of the gradients, hence we have the following lemma:

**Lemma 3.1.** The rotation matrix $A$ identified by eigen-decomposition of $G$ in Eq. \text{20} is the limit of $A$ identified by SVD of $W$ in Eq. \text{19}.

**Proof.** Apparently,

$$G = \lim_{M \to \infty} \frac{1}{M} W W^T.$$

Hence we have

$$G = \lim_{M \to \infty} \frac{1}{M} (U_W \Sigma_W V_W^T) \left( U_W \Sigma_W V_W^T \right)^T = \lim_{M \to \infty} \frac{1}{M} U_W \Sigma_W \Sigma_W^T U_W^T$$

(22)

Therefore, $U$ and $\Lambda$ are limits of $U_W$ and $\Sigma_W \Sigma_W^T/M$, respectively. \hfill \Box

The aforementioned selection of $A$ helps to concentrate the dependence of $u$ on fewer $\eta_i$ if the gaps between $\sigma_i$ (or equivalently the gaps between $\lambda_i$) are large. Therefore, it helps to yield a sparser $\tilde{c}$. We note that the idea of active subspace is similar to the method of outer product gradient (OPG) \cite{18, 40} and factor analysis \cite{15} in statistics.

Since $u$ is not known, we use $u_g$ to replace it in Eq. \text{18}:

$$W \approx W_g = [\nabla u_g(\xi^1), \nabla u_g(\xi^2), \cdots, \nabla u_g(\xi^M)],$$

(23)

and the rotation matrix is selected based on the SVD of $W_g$:

$$W_g = U_{W_g} \Sigma_{W_g} V_{W_g}^T, \quad A = U_{W_g}^T,$$

(24)
Here \( u_g \) can be computed with different methods, and we use standard \( \ell_1 \) (or reweighted \( \ell_1 \)) minimization to compute it. On obtaining \( A \), we define \( \eta = A\xi \) and compute the corresponding input samples as \( \eta^g = A\xi^g \). Then we construct a new measurement matrix \( \Psi(\eta) \) as \( (\Psi(\eta))_{ij} = \psi_j(\eta^g) \). Next, we solve the \( \ell_1 \) minimization problem \((P_{1,c})\) to obtain \( \hat{c} \).

### 3.2. Iterative method.

The improvement of the sparsity in \( u_g \) has the potential to leverage the compressive sensing algorithms to obtain more accurate results from limited data. Now we modify the \( \ell_1 \) minimization problem \((P_{1,c})\) as

\[
(P_{1,c}^R) \quad \underset{\hat{c}}{\arg\min} \| \hat{c} \|_1, \quad \text{subject to} \quad \| \Psi \hat{c} - u \|_2 \leq \varepsilon, \tag{25}
\]

where \( \Psi_{ij} = \psi_j(A\xi^g) \). The rotation matrix \( A \) obtained in Eq. (24) may not be a good answer to this optimization problem. This is because if we don’t have sufficient data, the accuracy of \( u_g \) can be low even if we use compressive sensing method, as such the SVD of \( W_g \) may be less accurate. Therefore, we propose an alternating direction type of iterative method to identify a better rotation matrix \( A \).

We start with an initial guess \( u_g \), which is obtained by standard \( \ell_1 \) minimization in this work. We denote its coefficients as \( c^{(0)} \), where the superscript \( \cdot^{(0)} \) stands for the \( l \)-th iteration. We also set \( A^{(0)} = I \), \( \eta^{(0)} = \xi \), \( \psi_j^{(0)}(\eta^{(0)}) = u_g(\xi) \). In the \( l \)-th iteration \((l \geq 1)\), given \( v^{(l-1)}_g \) and input samples \((\eta^{(l-1)})^g, q = 1, \ldots, M \), we first collect the gradient of \( v^{(l-1)}_g \):

\[
W^{(l-1)}_g = \begin{bmatrix}
\nabla_{\xi} v^{(l-1)}_g \left( (\eta^{(l-1)})^1 \right), \nabla_{\xi} v^{(l-1)}_g \left( (\eta^{(l-1)})^2 \right), \ldots, \nabla_{\xi} v^{(l-1)}_g \left( (\eta^{(l-1)})^M \right)
\end{bmatrix}, \tag{26}
\]

where \( \nabla_{\xi} = (\partial / \partial \xi_1, \partial / \partial \xi_2, \ldots, \partial / \partial \xi_d)^T \). Then we compute the SVD of \( W^{(l-1)}_g \):

\[
W^{(l-1)}_g = U^{(l-1)}_{W_g} \Sigma^{(l-1)}_{W_g} \left( V^{(l-1)}_{W_g} \right)^T, \tag{27}
\]

and set \( A^{(l)} = \left(U^{(l-1)}_{W_g} \right)^T \). Now we define a new set of random variables as \( \eta^{(l)} = A^{(l)}\xi \) and compute their samples accordingly: \( (\eta^{(l)})^g = A^{(l)} \xi^g \). Then we construct a new measurement matrix \( \Psi^{(l)} \) as \( \psi_j^{(l)}((\eta^{(l)})^g) \), and solve the \( \ell_1 \) minimization problem \((P_{1,c})\) to obtain \( \hat{c}^{(l)} \).

We note that the gradient in Eq. (26) is computed with respect to \( \xi \). In practice, we use chain rule in the computing:

\[
\nabla_{\xi} v^{(l)}_g \left( (\eta^{(l)})^g \right) = \nabla_{\xi} v^{(l)}_g \left( A^{(l)} \xi^g \right) = (A^{(l)})^T \nabla v^{(l)}_g(\xi) \bigg|_{\xi=A^{(l)}\xi^g} = (A^{(l)})^T \sum_{n=1}^{N} \hat{c}^{(l)} \nabla \psi_n(\xi) \bigg|_{\xi=A^{(l)}\xi^g}. \tag{28}
\]

Therefore, we only need to evaluate \( \nabla \psi_n \) at \((\eta^{(l)})^g \). This is straightforward since \( \nabla \psi_n \) of most generalized polynomial used in uncertainty quantification is known, e.g., the derivative of Hermite polynomial, Legendre polynomial and Chebyshev polynomial. We summarize the entire procedure in Algorithm 3. The stopping criterion we use here is the difference between to successive rotation matrix \( A^{(l)} \) and \( A^{(l+1)} \). The threshold can be set as a fraction of the dimension, e.g., 0.1d. Alternatively, it can be set as \( l \leq l_{\text{max}} \), where \( l_{\text{max}} \) is maximum practical problem. This is because 1) to take advantage of the improvement of the sparsity, we do not need a very exact solution of the rotation; 2) the improvement of the sparsity is problem dependent (see the Examples in the \[4\]) and for many practical problems the improvement is not significant by using more iterations. In a word, we update \( c^{(l)} \) and \( A^{(l)} \) separately in each iteration, which is the spirit of the alternating direction method for the optimization problem.

An alternative design of the algorithm is to modify Eq. (26) as:

\[
\tilde{W}^{(l-1)}_g = \begin{bmatrix}
\nabla v^{(l-1)}_g \left( (\eta^{(l-1)})^1 \right), \nabla v^{(l-1)}_g \left( (\eta^{(l-1)})^2 \right), \ldots, \nabla v^{(l-1)}_g \left( (\eta^{(l-1)})^M \right)
\end{bmatrix}. \tag{29}
\]
Matrix-matrix multiplication when computing the gradient (see Eq. (28) and Eq. (30)). But algorithms are equivalent. In computation, the second version save the cost of computing a matrix ˜
We note that the stopping criterion in this version is based on the variables successively in each iteration. The entire procedure is summarized in Algorithm 4.

Does not need to be computed explicitly, and this alternative algorithm rotates the random difference between the results of these two different versions of algorithms. If the gradient  

\[ \eta \]

should be close to  

\[ \theta \]

in each iteration.

\[ \text{Algorithm 3 Alternating direction method of solving } (P_{1,e}) \text{ (version 1)} \]

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

\[ \text{arg} \min_{\hat{c}} \| \hat{c} \|_1, \text{ subject to } \| \Psi \hat{c} - u \|_2 \leq \epsilon. \]

5: Set counter \[ l = 0, \eta^{(0)} = \xi, \hat{c}^{(0)} = \hat{c} \] and \( A^{(0)} = I \).
6: \( l = l + 1 \). Construct \( W_g^{(l-1)} \) by computing \( \nabla_{\xi} v_g^{(l-1)} ((\eta^{(l-1)})^q) \) according to Eq. (28). Then compute SVD of \( W_g^{(l-1)} \):

\[ W_g^{(l-1)} = U_{W_g}^{(l-1)} \Sigma_{W_g}^{(l-1)} \left( V_{W_g}^{(l-1)} \right)^T, \]

7: Set \( A^{(l)} = \left( U_{W_g}^{(l-1)} \right)^T \) and set \( \eta^{(l)} = A^{(l)} \xi \), then compute samples \( (\eta^{(l)})^q = A^{(l)} \xi^q, q = 1, 2, \cdots, M \). Also, construct the new measurement matrix \( \Psi^{(l)} \) with \( \Psi_{ij} = \psi_j((\eta^{(l)})^i) \).
8: Solve the optimization problem \( (P_{1,e(l)}) \):

\[ \text{arg} \min_{\hat{c}} \| \hat{c} \|_1, \text{ subject to } \| \Psi^{(l)} \hat{c} - u \|_2 \leq \epsilon^{(l)}, \]

and set \( \hat{c}^{(l)} = \hat{c} \).
9: If \( \| A^{(l)} - A^{(l-1)} \|_2 < \theta \), where the threshold \( \theta \) is a positive real number, then stop. Otherwise, go to Step 6.
10: Construct gPC expansion as \( u(\xi) \approx u_g(\xi) = v_g(A^{(l)} \xi) = \sum_{n=1}^N \tilde{c}_n^{(l)} \psi_n(A^{(l)} \xi) \).

Then we compute the SVD of \( \tilde{W}_g^{(l-1)} \):

\[ \tilde{W}_g^{(l-1)} = \tilde{U}_{W_g}^{(l-1)} \tilde{\Sigma}_{W_g}^{(l-1)} \left( \tilde{V}_{W_g}^{(l-1)} \right)^T, \]

and set \( \eta^{(l)} = (\tilde{U}_{W_g}^{(l-1)})^T \eta^{(l-1)} \). We note that the rotation matrix \( A^{(l)} = \left( U_{W_g}^{(0)} U_{W_g}^{(2)} \cdots U_{W_g}^{(l-1)} \right)^T \) does not need to be computed explicitly, and this alternative algorithm rotates the random variables successively in each iteration. The entire procedure is summarized in Algorithm 3. We note that the stopping criterion in this version is based on the \( L_1 \) norm of the rotation matrix \( \tilde{U}^{(l-1)} \). If this matrix is “close” to identity matrix or permutation matrix, its \( L_1 \) norm should be close to \( d \), which is the dimension of the random variables. The threshold can also be set as a fraction of the dimension, e.g., 0.1d. According to our experience, there is no significant difference between the results of these two different versions of algorithms. If the gradient of \( u \) can be computed accurately (although this is usually impossible in practice), these two algorithms are equivalent. In computation, the second version save the cost of computing a matrix-matrix multiplication when computing the gradient (see Eq. (28) and Eq. (31)). But this cost is trivial since the sizes of these matrices are \( d \times d \) and \( d \times M \). In this work, we use the first version in the numerical examples. Namely, the rotation matrix \( A^{(l)} \) is computed explicitly in each iteration.
3.3. A special case. It is straightforward to demonstrate that the Algorithm\textsuperscript{5}  for Hermite polynomial expansion (when \(\xi_i\) are i.i.d. Gaussian) is a special case of the Algorithm\textsuperscript{6} The only difference between Algorithms\textsuperscript{5} and\textsuperscript{6} is the approach of obtaining rotation matrix for rotating \(\eta^{(l-1)}\) to \(\eta^{(l)}\). Specifically, Algorithm\textsuperscript{5} uses \(\mathbf{U}_n\) in Eq. (13) while Algorithm\textsuperscript{6} uses \(\mathbf{U}\) in Eq. (21). We already demonstrat that asymptotically (i.e., when \(M \rightarrow \infty\)), matrix \(\mathbf{U}\) from eigen-decomposition of \(\mathbf{G}\) and \(\mathbf{U}_n\) from SVD of \(\mathbf{W}\) are the same. In practice \(u\) is approximated by \(u_\eta\) and and \(\mathbf{G}\) is approximated by an explicit form:

\[
\mathbf{G} \approx \mathbb{E} \{\nabla u_\eta(\xi) \otimes \nabla u_\eta(\xi)\} = \mathbb{E} \left\{ \nabla \left( \sum_{n=1}^{N} c_n \psi_n(\xi) \right) \otimes \nabla \left( \sum_{n'=1}^{N} c_{n'} \psi_{n'}(\xi) \right) \right\}, \tag{32}
\]

For simplicity, we use the following notation \(\partial_i h(x) = \frac{\partial h(x)}{\partial x_i}\) to denote the partial differential of a differentiable function \(h\). Then Eq. (32) implies that

\[
G_{ij} \approx \mathbb{E} \left\{ \partial_i \left( \sum_{n=1}^{N} c_n \psi_n(\xi) \right) \cdot \partial_j \left( \sum_{n'=1}^{N} c_{n'} \psi_{n'}(\xi) \right) \right\} = \mathbb{E} \left\{ \left( \sum_{n=1}^{N} c_n \partial_i \psi_n(\xi) \right) \cdot \left( \sum_{n'=1}^{N} c_{n'} \partial_j \psi_{n'}(\xi) \right) \right\} = \sum_{n=1}^{N} \sum_{n'=1}^{N} c_n c_{n'} \mathbb{E} \{ \partial_i \psi_n(\xi) \cdot \partial_j \psi_{n'}(\xi) \} = \mathbf{c}^T \mathbf{K}_{ij} \mathbf{c}, \tag{33}
\]
Notice that and mial expansion takes this advantage. The design of the algorithms for Hermite polynomials, eigen-decomposition of $G$, and construct $gPC$ expansion as $u_l(x) = \sum_{n=1}^{N} c_n^{(l)} \psi_n(A^{(l)} x)$. The design of the algorithms for Hermite polynomial expansion takes this advantage.

The design of the algorithms for Hermite polynomial expansion takes this advantage. The design of the algorithms for Hermite polynomial expansion takes this advantage.

\begin{algorithm}
\textbf{Algorithm 5 Alternating direction method of solving ($P_{1,e}^R$) when $\xi$ are i.i.d. Gaussian.}
\begin{enumerate}
\item Generate input samples $\xi^q, q = 1, 2, \cdots, M$ based on the distribution of $\xi$.
\item Generate output samples $u^q = u(\xi^q)$ by solving the complete model; e.g., running simulations, solvers, etc.
\item Select $gPC$ basis functions $\{\psi_n\}_{n=1}^N$ associated with $\xi$ and then generate the measurement matrix $\Psi$ by setting $\Psi_{ij} = \psi_j(\xi^i)$.
\item Solve the optimization problem ($P_{1,e}^R$):
\[
\arg\min_{\hat{c}} \|\hat{c}\|_1, \text{ subject to } \|\Psi \hat{c} - u\|_2 \leq \epsilon.
\]
\item Set counter $l = 0$, $\eta^{(0)} = \xi$, $\hat{c}^{(0)} = \hat{c}$, compute $K_{ij}, i, j = 1, 2, \cdots, N$.
\item $l = l + 1$. Construct $G^{(l-1)}$ as $G^{(l-1)}_{ij} = (\hat{c}^{(l-1)})^T K_{ij} \hat{c}^{(l-1)}$, $i, j = 1, 2, \cdots, d$. Then compute eigen-decomposition of $G^{(l-1)}$:
\[
G^{(l-1)} = U^{(l-1)} A^{(l-1)} \left(U^{(l-1)}\right)^T,
\]
\item Set $\eta^{(l)} = (U^{(l-1)})^T \eta^{(l-1)}$, then compute samples $(\eta^{(l)})^q = (U^{(l-1)})^T (\eta^{(l-1)})^q$, $q = 1, 2, \cdots, M$. Also, construct the new measurement matrix $\Psi^{(l)}$ with $\Psi^{(l)}_{ij} = \psi_j((\eta^{(l)})^q)$.
\item Solve the optimization problem ($P_{1,e}^{(l)}$):
\[
\arg\min_{\tilde{c}} \|\tilde{c}\|_1, \text{ subject to } \|\Psi^{(l)} \tilde{c} - u\|_2 \leq \epsilon^{(l)},
\]
and set $\tilde{c}^{(l)} = \tilde{c}$.
\item If $\|U^{(l-1)}\|_1 - d < \theta$, where the threshold $\theta$ is a positive real number, then stop. Otherwise, go to Step 6.
\item Set $A^{(l)} = \left(U^{(0)} U^{(2)} \cdots U^{(l-1)}\right)^T$, and construct $gPC$ expansion as $u_l(x) \approx u_g(x) = v_g(\xi) = \sum_{n=1}^{N} \tilde{c}_n^{(l)} \psi_n(A^{(l)} x)$.\end{enumerate}
\end{algorithm}

where $K_{ij}$ is a matrix defined as
\[
(K_{ij})_{km} = \mathbb{E}\left\{ \partial_i \psi_k(\xi) \cdot \partial_j \psi_m(\xi) \right\}. \tag{34}
\]
In each iteration, we do not need to update $K_{ij}$ because $\eta^{(l)}_i$ are i.i.d. Gaussian, and
\[
\mathbb{E}\left\{ \partial_i \psi_k(\eta^{(l)}) \cdot \partial_j \psi_m(\eta^{(l)}) \right\} = \frac{(2\pi)^{d/2}}{\sqrt{\det(D\eta^{(l)})}} \int_{\mathbb{R}^d} \partial_i \psi_k(x) \cdot \partial_j \psi_m(x) \exp\left(-\frac{\|x\|^2}{2}\right) \, dx
\]
are fixed so it can be precomputed (see [47]). The design of the algorithms for Hermite polynomial expansion takes this advantage.

Next we show that using the stiffness matrix $G$ based on matrices $K_{ij}$ to identify rotations may not be efficient for other types of polynomial expansions. In the $l$-th iteration,
\[
G^{(l-1)}_{ij} = \mathbb{E}\left\{ \partial_i \left( \sum_{n=1}^{N} \tilde{c}_n^{(l-1)} \psi_n(\eta^{(l-1)}) \right) \cdot \partial_j \left( \sum_{n'=1}^{N} \tilde{c}_{n'}^{(l-1)} \psi_{n'}(\eta^{(l-1)}) \right) \right\} = (\hat{c}^{(l-1)})^T K_{ij}^{(l-1)} \hat{c}^{(l-1)},
\]
and
\[
(K_{ij}^{(l-1)})_{km} = \mathbb{E}\left\{ \partial_i \psi_k(\eta^{(l-1)}) \cdot \partial_j \psi_m(\eta^{(l-1)}) \right\}. \tag{35}
\]
Notice that
\[
\left| \det \left( \frac{D\eta^{(l-1)}}{D\xi} \right) \right| = \left| \det \left( (A^{(l-1)})^{-1} \right) \right| = 1, \tag{36}
\]
hence, the PDF of $\eta^{(l-1)}$ is

$$
\rho_{\eta^{(l-1)}}(x) = \rho_x((A^{(l-1)})^{-1}x) \left| \det \left( \frac{D\xi}{D\eta^{(l-1)}} \right) \right| = \rho_x \left( (A^{(l-1)})^{-1}x \right).
$$

(37)

Thus, for given indices $n$ and $n'$, we have

$$
E \left\{ \partial_i \psi_n(\eta^{(l-1)}) \cdot \partial_j \psi_{n'}(\eta^{(l-1)}) \right\} = \int_{\Omega^{(l-1)}} \partial_i \psi_n(x) \cdot \partial_j \psi_{n'}(x) \rho_{\eta^{(l-1)}}(x) dx
$$

$$
= \int_{\Omega^{(l-1)}} \partial_i \psi_n(x) \cdot \partial_j \psi_{n'}(x) \rho_x((A^{(l-1)})^{-1}x) dx,
$$

(38)

where $\Omega^{(l-1)}$ is the domain of multivariate random variable $\eta^{(l-1)}$. If $\xi \sim N(0, I)$, then $\Omega^{(l-1)} = \Omega$ and

$$
\rho_{\eta^{(l-1)}}(x) = \rho_x((A^{(l-1)})^{-1}x) = \rho_x(x),
$$

(39)

since $A^{(l-1)}(A^{(l-1)})^T = I$. Hence, we only need to compute $K_{ij}$ once. In general cases, updating $K_{ij}$ in each iteration can be costly. A possible solution is to compute $K_{ij}$ based on $\rho_x$ first. Then, in each iteration, after obtaining $v_g^{(l)}(\eta^{(l)}) = \sum_{i=1}^{N} c_n^{(l)} \psi_n(\eta^{(l)})$, we compute the corresponding $u_g^{(l)}(\xi) = \sum_{i=1}^{N} c_n^{(l)} \psi_n(\xi)$ through algebraic computing or by accurate numerical integral:

$$
c_n^{(l)} = \int_{\Omega} v_g^{(l)}(A^{(l)}x) \psi_n(x) \rho_x(x) dx = \sum_{q=1}^{N_q} v_g^{(l)}(A^{(l)}x^q) \psi_n(x^q) w^q,
$$

where $x^q$ and $w^q$ are quadrature points and weights with respect to $\rho_x(x)$. As such, $G_{ij}^{(l)}$ can be approximated by $(c^l)^T K_{ij} c^{(l)}$. However, the additional computation cost for converting $v_g^{(l)}$ to $u_g^{(l)}$ makes this algorithm less efficient. According to our experience, there is not significant difference between the accuracy of Algorithms 4 and 5 for Hermite polynomial expansions.

3.4. Compromise of the property of $\Psi$. We discuss above the possible enhancement of the sparsity in $\tilde{c}$ by introducing the rotation. However, as we pointed out at the beginning of this section, $\{\psi_i\}_{i=1}^{N}$ are not necessarily orthonormal to each other with respect to $\rho_\eta$, the property of matrix $\Psi^{(l)}$ may become less favorite for the $\ell_1$ minimization. The most straightforward conclusion we can obtain is that the mutual coherence of $\Psi^{(l)}$ $(l \geq 1)$ is larger than that of $\Psi^{(0)}$. Only when $\psi_j$ are Hermite polynomials (i.e., $\{\xi_i\}_{i=1}^{d}$ are i.i.d. Gaussian), the property of $\Psi$ (e.g., mutual coherence) is conserved (statistically) since $\eta^{(l)}$ are still i.i.d. Gaussian. This adds to the specialty of the Hermite polynomial. For other polynomials, the aforementioned rotational method can be less efficient in some cases due to the compromise of the property of $\Psi$ as we will see in the next section. We note that in the compressive sensing theory, the number of samples needed for an accurate computing of $c$ is related to both the sparsity of $c$ and the property of $\Psi$. Therefore, for general cases, even though we can increase the sparsity of $c$, if $\Psi$ becomes worse (i.e., it loses RIP or the mutual coherence increases), the efficiency of our method can be affected.

4. Numerical Examples

In this section, we revisit the five numerical examples in [17] with different types of random variables in the systems and with different types of polynomial expansions to approximate the solution. By testing the same examples (with different types of random variables), we demonstrate the efficiency of the proposed general framework and we can compare the performance with the special case (Hermite polynomial expansion) in our previous study. Specifically, the random variables we consider in this section are uniform random variables $U[-1, 1]^d$ (associated with Legendre polynomials) and Chebyshev random variables with PDF $\rho(x) = \left( \frac{1}{\pi \sqrt{1 - x^2}} \right)^d$.
in our figures as the ratio $M/N$. Therefore, as we keep the set of the basis functions unchanged, all the polynomials not related where all $\xi$ samples. To investigate the effectiveness of the increasing of output samples, we set the model hierarchically based on the importance of $\xi$ unknowns. We use MATLAB package $\xi$ polynomial expansion (assuming we use Algorithm 3 and results are obtained with $l$ polynomial expansion (assuming $\xi$ grids method, hence they are exact. The relative errors are presented in Fig. 1 for Legendre $A$-minimization. The integrals for calculating the $L$ Ridge function. Consider the following ridge function

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

(41)

where all $\xi_i$ are equally important. In this case, adaptive methods that build the surrogate model hierarchically based on the importance of $\xi_i$ (e.g., [25, 45, 50]) may not be efficient. A simple rotation matrix for this example has the form

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

(42)

where $\hat{A}$ 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:

$$u(\xi) = u(\eta) = d^{1/2} \eta_1 + 0.25 d \eta_2^2 + 0.025 d^{3/2} \eta_3^3.$$

Therefore, as we keep the set of the basis functions unchanged, all the polynomials not related to $\eta_1$ make no contribution to the expansion, which implies that we obtain a very sparse representation of $u$. Since the optimal structure is not known a priori, 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 grids method, hence they are exact. The relative errors are presented in Fig. 1 for Legendre polynomial expansion (assuming $\xi_i$ are i.i.d. uniform random variables) and Chebyshev polynomial expansion (assuming $\xi_i$ are i.i.d. Chebyshev random variables). Clearly, the standard $\ell_1$ minimization is not effective as the relative error is close to 50% even when $M/N$ is close to 0.4. Also, the re-weighted $\ell_1$ does not help in this case. However, our new iterative rotation demonstrates much better accuracy, especially when $M$ is large.

Fig. 2 compares the absolute values of exact coefficients $c_n$ and the coefficients $\tilde{c}_n$ after 9 iterations using 180 samples. We note that in this figure we do not present $\tilde{c}_n$ with absolute value smaller than $10^{-3}$ since they are more than two magnitude smaller than the dominating ones. As demonstrated in Fig. 2 the iterative rotation creates a much sparser representation

(associate with Chebyshev polynomials of the first kind). For simplicity, we use Chebyshev polynomial to denote Chebyshev polynomial of the first kind. The accuracies of different methods are measured by the relative $L_2$ error: $(\|u - u_g\|_2)/\|u\|_2$, where $u_g$ is the Legendre polynomial expansion or Chebyshev polynomial expansion of $u$. The integral

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

(40)

(and $\|u - u_g\|_2$) is approximated with a high-level sparse grids method which is based on one-dimensional Gauss quadrature and the Smolyak structure [35]. The term “level” $p$ means that the algebraic accuracy of the sparse grids method is $2p - 1$. We use $P$ to denote the truncation order, which implies that polynomials up to order $P$ are included in expansion $u_g$. Hence, the number of unknowns can be computed as $N = (P+d)\text{max}$. The relative errors we present in this section are obtained from 50 independent replicates for each sample size $M$. For example, we generate 50 independent sets of input samples $\xi^q, q = 1, 2, \cdots, M$, compute 50 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 [39, 38] to solve ($P_i, q$). If not otherwise indicated, we use Algorithm 4 and results are obtained with $l_{\text{max}} = 3$ iterations and we use three iterations in the re-weighted $\ell_1$ minimization.

4.1. Ridge function. Consider the following ridge function
of $u$, hence the efficiency of compressive sensing method is substantially enhanced. We notice that the accuracy increases as more iterations are included. Moreover, the improvement from 6 iterations to 9 iterations is less significant as that from 3 iterations to 6 iterations. We note that this is a special example in that ridge function has very good low-dimensional structure (it is a one-dimensional function after an appropriate linear transform). In general, many systems does not have this ideal structure and the improvement afforded by iterative rotation usually becomes very small after 2 to 3 iterations.

4.2. Function with high compressibility. Consider the following function:

$$u(\xi) = \sum_{|\alpha|=0}^{P} c_\alpha \psi_\alpha(\xi) = \sum_{n=1}^{N} c_n \psi_n(\xi), \quad \xi = (\xi_1, \xi_2, \cdots, \xi_d),$$

where, $\psi_\alpha$ are normalized multivariate Legendre polynomials or Chebyshev polynomials, $d = 12$, $P = 3$, $N = 455$, and the coefficients $c_n$ are chosen as uniformly distributed random numbers, $c_n = \zeta/n^{1.5}$, $\zeta \sim U[0, 1]$. (44)

For this example, we generate $N$ samples of $\zeta$: $\zeta_1, \zeta_2, \cdots, \zeta_N$ 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. Figure 3 shows the relative $L_2$ errors obtained by applying our iterative rotation technique to the re-weighted $\ell_1$ approach. Apparently, introduction of the iterative rotation approach improves the accuracy. A comparison of the absolute values of entries of $c$ and $\tilde{c}$ (using 160 samples) is presented in Fig. 4. The main improvement is that the number of coefficients with magnitude larger than 0.01 is decreased. Also, $c_n$ cluster around the curve $c_n = 1/n^{1.5}$ as we set them in this way, while many $\tilde{c}_n$ are much below this curve especially when $n$ is large. We also compare the values of $\|c - c_s\|_1 / \sqrt{s}$ and $\|\tilde{c} - \tilde{c}_s\|_1 / \sqrt{s}$ in this figure to demonstrate the enhancement of the sparsity after rotations quantitatively.

4.3. Elliptic equation. Next we consider a one-dimensional elliptic differential equation with a random coefficient:

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

$$u(0) = u(1) = 0,$$

(45)
Figure 2. Results for the ridge function. Absolute values of exact coefficients $c_n$ and coefficients $\tilde{c}_n$ after rotations using 180 samples.

Figure 3. Results for the highly compressible function. Left: Legendre polynomial expansion (when $\xi_i$ are i.i.d. uniform random variables); Right: Chebyshev polynomial expansion (when $\xi_i$ are i.i.d. Chebyshev random variables). “o”: standard $\ell_1$, “∗”: re-weighted $\ell_1$, “△”: rotated $\ell_1$, “♦”: re-weighted+rotated $\ell_1$.

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

$$a(x; \xi) = a_0(x) + \exp \left( \sigma \sum_{i=1}^{d} \sqrt{\lambda_i} \phi_i(x) \xi_i \right), \quad (46)$$
where \( \{\xi_i\} \) are i.i.d. 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:

\[
C(x, x') = \exp \left( \frac{|x - x'|}{l_c} \right).
\]

In the KL expansion, \( \lambda_i \) denotes the eigenvalue of the covariance kernel \( C(x, x') \) instead of entries of \( \Lambda \) in Eq. (33). The value of \( \lambda_i \) and the analytical expressions for \( \phi_i \) are available in the literature [20]. 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 \( \xi^q \), \( a \) and \( u \) only depend on \( x \) and the solution of the deterministic elliptic equation can be obtained as [46]:

\[
u(x) = u(0) + \int_{0}^{x} a(0)u(0)' - y \frac{a(y)}{a(y)} dy.
\]

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

\[
a(0)u(0)' = \left( \int_{0}^{1} \frac{y}{a(y)} dy \right) / \left( \int_{0}^{1} \frac{1}{a(y)} dy \right).
\]

The integrals in Eqs. (49) and (48) are obtained by highly accurate numerical integration. For this example, we choose the quantity of interest to be \( u(x; \xi) \) at \( x = 0.35 \). We aim to build a 3rd-order Legendre (or Chebyshev) polynomial expansion which includes \( N = 816 \) basis functions. The relative error is approximated by a level-6 sparse grid method. Figure 5 shows that accuracy of the re-weighted \( \ell_1 \) and the iteratively rotated \( \ell_1 \) method are very close in this case. In the Legendre polynomial expansion, the incorporation of iterative rotation improves the performance of the other methods. In the Chebyshev polynomial expansion, the improvement

\[\text{Figure 4. Results for the highly compressible function. Left column: absolute values of exact coefficients } c_n; \text{ middle column: absolute values of coefficients } \tilde{c}_n \text{ after rotations using 160 samples; right column: comparison of } \|c - c_s\|_1 / \sqrt{s} ("\circ") \text{ and } \|\tilde{c} - \tilde{c}_s\|_1 / \sqrt{s} ("\diamond") \text{ with different } s.\]
is very little. This is related to the compromise of the property of \( \Psi \). A comparison of \( c \) and \( \tilde{c} \) are presented in Fig. 6, which shows the improvement of the sparsity in the similar manner as in function with high compressibility in Sec. 4.2. Here we plot coefficients with absolute values larger than \( 10^{-8} \) for demonstration purpose since other entries are negligible in the comparison of sparsity. The enhancement of sparsity is also illustrated quantitatively in this figure on the right column.

**Figure 5.** Results for the elliptic equation. Left: Legendre polynomial expansion (when \( \xi_i \) are i.i.d. uniform random variables); Right: Chebyshev polynomial expansion (when \( \xi_i \) are i.i.d. Chebyshev random variables). “◦”: standard \( \ell_1 \), “*: re-weighted \( \ell_1 \), “⊲”: rotated \( \ell_1 \), “⋄”: re-weighted+rotated \( \ell_1 \).

**Figure 6.** Results for the elliptic equation. Left column: absolute value of exact coefficients \( |c_n| \); middle column: absolute values of coefficients \( \tilde{c}_n \) after rotations using 160 samples; right column: comparison of \( \frac{||c - c_s||_1}{\sqrt{s}} \) (“◦”) and \( \frac{||\tilde{c} - \tilde{c}_s||_1}{\sqrt{s}} \) (“⋄”) with different \( s \).
4.4. 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 [24]:

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

\[ u(x, 0; \xi) = -2 \text{sech}^2(x). \]  

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

\[ f(t; \xi) = \sigma \sum_{i=1}^{d} \sqrt{\lambda_i} \phi_i(t) \xi_i, \]  

where \( \sigma \) is a constant and \( \{\lambda_i, \phi_i(t)\}_{i=1}^{d} \) are eigenpairs of the exponential covariance kernel as in Eqs. (16) and (17), 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

\[ u(x, t; \xi) = \sigma \sum_{i=1}^{d} \sqrt{\lambda_i} \xi_i \int_0^t \phi_i(y)dy - 2 \text{sech}^2 \left( x - 4t + 6\sigma \sum_{i=1}^{d} \sqrt{\lambda_i} \xi_i \int_0^t \int_0^\infty \phi_i(y)dydz \right). \]  

The quantity of interest is chosen to be \( u(x, t; \xi) \) at \( x = 6, t = 1 \) with \( \sigma = 0.4 \). Since an analytical expression for \( \phi_i \) is available, we can compute the integrals in Eq. (52) with high accuracy. Denoting

\[ A_i = \sqrt{\lambda_i} \int_0^1 \phi_i(y)dy, \quad B_i = \sqrt{\lambda_i} \int_0^1 \int_0^\infty \phi_i(y)dydz, \quad i = 1, 2, \cdots, d, \]

the analytical solution is

\[ u(x, t; \xi) |_{x=6,t=1} = \sigma \sum_{i=1}^{d} A_i \xi_i - 2 \text{sech}^2 \left( 2 + 6\sigma \sum_{i=1}^{d} B_i \xi_i \right). \]  

We use a fourth-order gPC expansion to approximate the solution, i.e., \( P = 4 \), and the number of gPC basis functions \( N = 1001 \). The \( L_2 \) error of Legendre polynomial expansion and Chebyshev polynomial expansion are presented in Fig. 7. For this example, the combined iterative rotation and re-weighted \( \ell_1 \) method outperforms all other approaches. However, in the Chebyshev polynomial expansion, when the sample size is small (i.e., \( M/N < 0.12 \)) if we only use the rotational method, the result is not as good as that by the standard \( \ell_1 \) minimization. This phenomenon is also related to the compromise in the property of \( \Psi \). A comparison of \( c \) and \( \tilde{c} \) (obtained using 180 samples) are presented in Fig. 8, which shows the improvement of the sparsity by the iterative rotation method. Coefficients with absolute values smaller than \( 10^{-8} \) are not presented since they are negligible in the comparison of sparsity. We also demonstrate the enhancement of the sparsity quantitatively in the figure.

4.5. High-dimensional function. In the last example, we illustrate the potential capability of the rotational method for dealing with higher-dimensional problems. Specially, we select a function similar to the first example (Sec. 4.1) but with much higher dimensionality:

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

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. 9. As before, our iterative rotation approach out-performs the existing \( \ell_1 \) methods. A comparison of \( c \) and \( \tilde{c} \) is presented in Fig. 10 and it shows the enhancement of the sparsity by the iterative rotation method. Coefficients \( \tilde{c}_n \) (obtained with 1200 samples) with absolute values smaller than \( 10^{-4} \) are not presented as they are two magnitude smaller than the dominating ones and are negligible in the comparison of sparsity. The enhancement of the sparsity is illustrated quantitatively on the right column of this figure. We note that
Figure 7. Results for the Korteweg-de Vries equation. Left: Legendre polynomial expansion (when $\xi_i$ are i.i.d. uniform random variables); Right: Chebyshev polynomial expansion (when $\xi_i$ are i.i.d. Chebyshev random variables). "◦": standard $\ell_1$, "∗": re-weighted $\ell_1$, "⊲": rotated $\ell_1$, "⋄": re-weighted+rotated $\ell_1$.

Figure 8. Results for the KdV equation. Left column: absolute values of exact coefficients $c_n$; middle column: absolute values coefficients $\tilde{c}_n$ after rotations using 180 samples; right column: comparison of $\|c - c_s\|_1 / \sqrt{s}$ ("◦") and $\|\tilde{c} - \tilde{c}_s\|_1 / \sqrt{s}$ ("⋄") with different $s$.

for general high-dimensional problems, simply truncating the gPC expansion up to a certain order is not efficient because the number of basis grows exponentially. Hence, a good approach for high-dimensional problems is to integrate our iterative rotation method with a method to reduce $d$ (e.g., an ANOVA method [45] or to reduce $N$ (e.g., adaptive basis selection [19].

4.6. Increase in the mutual coherence. As we point out in Sec. [3.4] the property of $\Psi^{(l)}$ becomes less favorite for $\ell_1$ minimization as we use Legendre and Chebyshev polynomials in
Figure 9. Results for the high-dimensional function. Left: Legendre polynomial expansion (when $\xi_i$ are i.i.d. uniform random variables); Right: Chebyshev polynomial expansion (when $\xi_i$ are i.i.d. Chebyshev random variables). “◦”: standard $\ell_1$, “∗”: re-weighted $\ell_1$, “⊲”: rotated $\ell_1$, “⋄”: re-weighted+rotated $\ell_1$.

Figure 10. Results for the high-dimensional function. Left column: absolute values of exact coefficients $c_n$; middle column: absolute values of coefficients $\tilde{c}_n$ after rotations using 1200 samples; right column: comparison of $\|c - c_s\|_1 / \sqrt{s}$ (“◦”) and $\|\tilde{c} - \tilde{c}_s\|_1 / \sqrt{s}$ (“○”) with different $s$.

The expansion. Here we use the mutual coherence $\mu(\Psi)$ (see Eq. (15)) to demonstrate this phenomenon. We use the set up of ridge function in example 1 ($d = 12, P = 3, N = 455$) and we use the exact rotation matrix $A$ in Eq. (42) to illustrate how the mutual coherence changes. The matrix $\Psi$ after rotation is computed as $\Psi_{ij} = \psi_j(A\xi^i)$, where $A$ is given in Eq. (42). We repeat the computing of $\mu(\Psi)$ with 50 independent sets of $\{\xi^i\}_{i=1}^{180}$ and present the average value in Table 4. As we anticipate, for Hermite polynomial, $\mu$ does not change, while for other type of
polynomials, it increases. Further, the increase in the Chebyshev polynomial is larger than that of the Legendre polynomials. This also provide a partial explanation why in some of our test cases, the rotational method for Chebyshev polynomial is less efficient than for the Legendre polynomial. Although theoretical analysis is not available at this time, this table provides an intuitive understanding of the algorithm.

|                | Legendre | Chebyshev | Hermite |
|----------------|----------|-----------|---------|
| $\mu(\Psi)$ before rotation | 0.15     | 0.15      | 0.40    |
| $\mu(\Psi)$ after rotation   | 0.45     | 0.50      | 0.40    |

5. Conclusions

In this paper, we provide a general framework of enhancing sparsity of gPC expansion by using alternating direction method to identify a rotation iteratively. As such, it improves the accuracy of compressive sensing method to construct the gPC expansions from a small size of data. The rotation is decided by seeking the directions of maximum variation for the QoI through SVD of the gradients at different points in the parameter space. We also demonstrate that our previously developed iterative method for Hermite polynomial expansion [47] is a special case of this general framework. Similar idea is also used in various methods, e.g., active subspace [32, 10], factor analysis [15], basis adaptation [37], etc. We note that unlike these methods, which identify the rotation matrix directly and truncate the dimension with possibly less accuracy detection of low-dimensional structure, our method does not truncate the dimension and identify the rotation matrix iteratively. This is because if we truncate the dimension with an inaccurate rotation matrix, the reduced model is less accurate. On the other hand, the dimension reduction can be performed after we identify accurately the important subspaces through the rotation. As such, these methods can be combined with our method.

In this work, we have demonstrated the works method for $\ell_1$ minimization but it can also be integrated with other optimization method to solve the compressive sensing problem, e.g., OMP [4], $\ell_{1-2}$ minimization [49], etc. In addition, future work will investigate the integration of our new methods with advanced sampling strategies (e.g., [17]), adaptive basis selection method (e.g., [19]), Bayesian compressive sensing method (e.g., [21]), etc. These advanced strategies are particularly important for high-dimensional problems. With these methods, we will be able to construct a more accurate surrogate model of the QoI with limited data. This is specifically useful for the problems where the experiments or simulations are costly, e.g., [34, 48]. This surrogate model can be used to study the sensitivity of the parameters and is very useful in inverse problems based on Bayesian framework.

Along with our previous work, we demonstrated the effectiveness of the rotational method for the Hermite polynomial expansions and Legendre polynomial expansions. These two are the most useful gPC expansions used in the uncertainty quantification study as the Gaussian and uniform random variables are most widely used in practice. Our method requires fewer samples of QoI to construct surrogate models, which can be a great savings of experimental or computational resources. As such, it is useful for most UQ problems. We note that the main limitation of applying our method to other types of gPC expansions is the possible compromise of the property of the measurement matrix. For example, for the Laguerre polynomial expansion, numerical tests (not presented in this work) show that our method does not work for some cases even if the coefficient has low dimensional structure. An intuitive guidance for using our method is that it works when the PDF is symmetric and, preferably, the measure concentrate near the origin.

Finally, we highlight two additional areas of future work for improving the new method. Firstly, the new method requires a formal numerical analysis to assess convergence behavior,
determine specific terminating criteria and the balance between the sparsity of coefficients and the property of measurement matrix. Second, there are likely more optimal iterative rotation strategies that can be applied. For example, it is possible that a non-linear transform may provide better sparse structure of the gPC expansion. Both of these questions will be addressed in our future work.

Acknowledgments

The research described in this paper was conducted under the Laboratory Directed Research and Development Program at Pacific Northwest National Laboratory, a multiprogram national laboratory operated by Battelle for the U.S. Department of Energy. Pacific Northwest National Laboratory is operated by Battelle for the DOE under Contract DE-AC05-76RL01830. A portion of this work is supported by the of Energy, Office of Science, Office of Advanced Scientific-Computing Research as part of the Collaboratory on Mathematics for Mesoscopic Modeling of Materials (CM4). We would like to thank Dr. Nathan Baker for fruitful discussions.

References

[1] Ben Adcock. Infinite-dimensional compressed sensing and function interpolation. *Foundations of Computational Mathematics*, pages 1–41, 2017.
[2] Negin Alemazkoor and Hadi Meidani. Divide and conquer: An incremental sparsity promoting compressive sampling approach for polynomial chaos expansions. *Computer Methods in Applied Mechanics and Engineering*, 318:937–956, 2017.
[3] Ivo Babuška, Fabio Nobile, and Raul Tempone. A stochastic collocation method for elliptic partial differential equations with random input data. *SIAM Rev.*, 52(2):317–355, 2010.
[4] Alfred M. Bruckstein, David L. Donoho, and Michael Elad. From sparse solutions of systems of equations to sparse modeling of signals and images. *SIAM Rev.*, 51(1):34–81, 2009.
[5] Robert H Cameron and William T Martin. The orthogonal development of non-linear functionals in series of fourier-hermite functionals. *Annals of Mathematics*, pages 385–392, 1947.
[6] Emmanuel J. Candès. The restricted isometry property and its implications for compressed sensing. *C. R. Math. Acad. Sci. Paris*, 346(9-10):589–592, 2008.
[7] Emmanuel J Candès, Justin K Romberg, and Terence Tao. Stable signal recovery from incomplete and inaccurate measurements. *Comm. Pure Applied Math.*, 59(8):1207–1223, 2006.
[8] Emmanuel J. Candès and Terence Tao. Decoding by linear programming. *IEEE Trans. Inform. Theory*, 51(12):4203–4215, 2005.
[9] Emmanuel J. Candès, Michael B. Wakin, and Stephen P. Boyd. Enhancing sparsity by reweighted $l_1$ minimization. *J. Fourier Anal. Appl.*, 14(5-6):877–905, 2008.
[10] David L Donoho. Compressed sensing. *IEEE Trans. Inf. Theory*, 52(4):1289–1306, 2006.
[11] David L. Donoho, Michael Elad, and Vladimir N. Temlyakov. Stable recovery of sparse overcomplete representations in the presence of noise. *IEEE Trans. Inform. Theory*, 52(1):6–18, 2006.
[12] Ali Doostan and Houman Owhadi. A non-adapted sparse approximation of PDEs with stochastic inputs. *J. Comput. Phys.*, 230(8):3015–3034, 2011.
[13] Jasmine Foo, Xiaoliang Wan, and George Em Karniadakis. The multi-element probabilistic collocation method (ME-PCM): error analysis and applications. *J. Comput. Phys.*, 227(22):9572–9595, 2008.
[14] Benjamin Fruchter. Introduction to factor analysis. 1954.
[15] Roger G. Ghanem and Pol D. Spanos. *Stochastic finite elements: a spectral approach*. Springer-Verlag, New York, 1991.
[16] Jerrad Hampton and Alireza Doostan. Compressive sampling of polynomial chaos expansions: Convergence analysis and sampling strategies. *J. Comput. Phys.*, 280(0):363 – 386, 2015.
[17] Georgios Karagiannis, Bledar A Konomi, and Guang Lin. A bayesian mixed shrinkage prior procedure for spatial–stochastic basis selection and evaluation of gpc expansions: Applications to elliptic spdes. *J. Comput. Phys.*, 284:528–546, 2015.
[23] Huan Lei, Xiu Yang, Bin Zheng, Guang Lin, and Nathan A Baker. Constructing surrogate models of complex systems with enhanced sparsity: quantifying the influence of conformational uncertainty in biomolecular solvation. SIAM Multiscale Model. Simul., 13(4):1327–1353, 2015.

[24] Guang Lin, Leopold Grinberg, and George Em Karniadakis. Numerical studies of the stochastic korteweg-de vries equation. J. Comput. Phys., 213(2):676–703, 2006.

[25] Xiang Ma and Nicholas Zabaras. An adaptive high-dimensional stochastic model representation technique for the solution of stochastic partial differential equations. J. Comput. Phys., 229(10):3884–3915, 2010.

[26] D. Needell. Noisy signal recovery via iterative reweighted $l_1$ minimization. In Proc. Asilomar Conf. on Signal Systems and Computers, Pacific Grove, CA, 2009.

[27] Hisanao Ogura. Orthogonal functionals of the poisson process. Information Theory, IEEE Transactions on, 18(4):473–481, 1972.

[28] Ji Peng, Jerrad Hampton, and Alireza Doostan. A weighted $l_1$-minimization approach for sparse polynomial chaos expansions. J. Comput. Phys., 267(0):92 – 111, 2014.

[29] Ji Peng, Jerrad Hampton, and Alireza Doostan. On polynomial chaos expansion via gradient-enhanced $l_1$-minimization. Journal of Computational Physics, 310:440 – 458, 2016.

[30] Holger Rauhut and Rachel Ward. Sparse legendre expansions via $l_1$-minimization. J. Approx. Theory, 164(5):517–533, May 2012.

[31] Holger Rauhut and Rachel Ward. Interpolation via weighted $l_1$ minimization. Applied and Computational Harmonic Analysis, 40(2):312 – 351, 2016.

[32] Trent Michael Russi. Uncertainty quantification with experimental data and complex system models. PhD thesis, UC Berkeley, 2010.

[33] Khachik Sargsyan, Cosmin Safta, Habib N Najm, Bert J Debusschere, Daniel Ricciuto, and Peter Thornton. Dimensionality reduction for complex models via bayesian compressive sensing. Int. J. Uncertainty Quantification, 4(1), 2014.

[34] Khachik Sargsyan, Cosmin Safta, Habib N. Najm, Bert J. Debusschere, Daniel Ricciuto, and Peter Thornton. Dimensionality reduction for complex models via bayesian compressive sensing. International Journal for Uncertainty Quantification, 4(1):63–93, 2014.

[35] S. Smolyak. Quadrature and interpolation formulas for tensor products of certain classes of functions. Sov. Math. Dokl., 4:240–243, 1963.

[36] Menner A Tatang, Wenwei Pan, Ronald G Prinn, and Gregory J McRae. An efficient method for parametric uncertainty analysis of numerical geophysical models. J. Geophys. Res-Atmos. (1984–2012), 102(D18):21925–21932, 1997.

[37] Ramakrishna Tipireddy and Roger Ghanem. Basis adaptation in homogeneous chaos spaces. Journal of Computational Physics, 259:304 – 317, 2014.

[38] E. van den Berg and M. P. Friedlander. SPGL1: A solver for large-scale sparse reconstruction, June 2007. http://www.cs.ubc.ca/labs/scl/spgl1.

[39] E. van den Berg and M. P. Friedlander. Probing the pareto frontier for basis pursuit solutions. SIAM Journal on Scientific Computing, 31(2):890–912, 2008.

[40] Yingcun Xia. A constructive approach to the estimation of dimension reduction directions. Ann. Stat., pages 2654–2690, 2007.

[41] Dongbin Xiu and Jan S. Hesthaven. High-order collocation methods for differential equations with random inputs. SIAM J. Sci. Comput., 27(3):1118–1139, 2005.

[42] Dongbin Xiu and George Em Karniadakis. The Wiener-Askey polynomial chaos for stochastic differential equations. SIAM J. Sci. Comput., 24(2):619–644, 2002.

[43] Zhiqiang Xu and Tao Zhou. On sparse interpolation and the design of deterministic interpolation points. SIAM J. Sci. Comput., 36(4):A1752–A1769, 2014.

[44] Liang Yan, Ling Guo, and Dongbin Xiu. Stochastic collocation algorithms using $l_1$-minimization. Int. J. Uncertainty Quantification, 2(3):279–293, 2012.

[45] Xiu Yang, Minseok Choi, Guang Lin, and George Em Karniadakis. Adaptive ANOVA decomposition of stochastic incompressible and compressible flows. J. Comput. Phys., 231(4):1587–1614, 2012.

[46] Xiu Yang and George Em Karniadakis. Reweighted $l_1$ minimization method for stochastic elliptic differential equations. J. Comput. Phys., 248(1):87–108, 2013.

[47] Xiu Yang, Huan Lei, Nathan Baker, and Guang Lin. Enhancing sparsity of hermite polynomial expansions by iterative rotations. J. Comput. Phys., 307:94–109, 2016.

[48] Xiu Yang, Huan Lei, Peiyuan Gao, Dennis G Thomas, David Mobley, and Nathan A Baker. Atomic radius and charge parameter uncertainty in biomolecular solvation energy calculations. arXiv preprint arXiv:1705.10035, 2017.

[49] Penghang Yin, Yifei Lou, Qi He, and Jack Xin. Minimization of 1-2 for compressed sensing. SIAM Journal on Scientific Computing, 37(1):A536–A563, 2015.

[50] Zheng Zhang, Xiu Yang, Ivan V Oseledets, George E Karniadakis, and Luca Daniel. Enabling high-dimensional hierarchical uncertainty quantification by anova and tensor-train decomposition. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 34(1):63–76, 2015.