Stochastic Collocation with Non-Gaussian Correlated Process Variations: Theory, Algorithms and Applications

Chunfeng Cui, and Zheng Zhang, *Member, IEEE*

***Abstract*—Stochastic spectral methods have achieved great success in the uncertainty quantification of many engineering problems, including variation-aware electronic and photonic design automation. State-of-the-art techniques employ generalized polynomial-chaos expansions and assume that all random parameters are independent or Gaussian correlated. This assumption is rarely true in real applications. How to handle non-Gaussian correlated random parameters is a long-standing and fundamental challenge: it is not clear how to choose basis functions and to perform a projection step in a correlated uncertain parameter space. This paper first presents a new set of basis functions to well capture the impact of non-Gaussian correlated parameters, then proposes an automatic and optimization-based quadrature method to perform projection-based stochastic collocation with a few simulation samples in the correlated parameter space. We further provide some theoretical proofs for the complexity and error bound of our proposed method. The numerical experiments on several synthetic, electronic and photonic integrated circuit examples show the nearly exponential convergence rate of our approach and its significant ($700\times$ to $6000\times$) speedup than Monte Carlo. Many other open problems with non-Gaussian correlated uncertainties can be further solved based on this work.**

***Index Terms*—Non-Gaussian correlation, uncertainty quantification, process variation, integrated circuits, integrated photonics, design automation algorithm.**

I. Introduction

PROCESS variation (e.g., random doping fluctuations and line edge roughness) is a major concern in nano-scale fabrications [2]: even a random difference on the atomic scale can have a large impact on the electrical properties of electronic integrated circuits (IC) [3], causing significant performance degradation and yield reduction. This issue is more severe in photonic IC [4]–[6], as photonic IC is much more sensitive to geometric variations such as surface roughness due to its large device dimension compared with the small operation wavelength [7]. In order to address this long-standing and increasingly important issue, efficient uncertainty quantification tools should be developed to predict and control the uncertainties of chip performance under various process variations. Due to its ease of implementation, Monte Carlo [7], [8] has been used in many commercial design automation tools. However, a Monte Carlo method often requires a huge number of device- or circuit-level simulation samples to achieve acceptable accuracy, and thus it is very time-consuming. As an alternative, stochastic spectral methods [9] may achieve orders-of-magnitude speedup over Monte Carlo methods in many application domains.

A stochastic spectral method approximates an unknown uncertain quantity (e.g., the nodal voltage, branch current or power dissipation of a circuit) as a linear combination of some specialized basis functions such as the generalized polynomial chaos [10]. Both intrusive (i.e., non-sampling) solvers (e.g., stochastic Galerkin [11] and stochastic testing [12]) and non-intrusive (i.e., sampling) solvers (e.g., stochastic collocation [13]) have been developed to compute the unknown weights of these pre-defined basis functions. These techniques have been successfully applied in electronic IC [14]–[23], MEMS [24], [25] and photonic IC [26], [27] applications, achieving orders-of-magnitude speedup than Monte Carlo when the number of random parameters is small or medium. In the past few years, there has been a rapid progress in developing high-dimensional uncertainty quantification solvers. Representative results include tensor recovery [28], compressive sensing [29], ANOVA (analysis of variance) or HDMR (high-dimensional model representation) [24], [30], [31], matrix low-rank approximation [32], stochastic model order reduction [33], and hierarchical uncertainty quantification [24], [25].

The above existing techniques use generalized polynomial chaos [10] as their basis functions, and they assume that all process variations can be described by independent random parameters. Unfortunately, this assumption is not true in many practical cases. For instance, the geometric or electrical parameters influenced by the same fabrication step are often highly correlated. In a system-level analysis, the performance parameters from circuit-level simulations are used as the inputs of a system-level simulator, and these circuit-level performance quantities usually depend on each other due to the network coupling and feedback. In photonic IC, spatial correlations may have to be considered for almost all components due to the small wavelength [34]. All these correlations are not guaranteed to be Gaussian, and they can not be handled by pre-processing techniques such as principal component analysis [35]. Karhunen-Loevè theorem [36], [37] and Rosenblatt transformation [38] may transform correlated parameters into uncorrelated ones, but they are error-prone and not scalable.

This paper develops new theory and algorithms of uncertainty quantification with non-Gaussian correlated process variations. Two main challenges arise when we quantify the

Some preliminary results of this work were reported in [1] and received the best conference paper award at EPEPS 2018. This work was partly supported by NSF-CCF Award No. 1763699, the UCSB start-up grant and a Samsung Gift Funding.

Chufeng Cui and Zheng Zhang are with the Department of Electrical and Computer Engineering, University of California, Santa Barbara, CA 93106, USA (e-mail: chunfengcui@ucsb.edu, zhengzhang@ece.ucsb.edu).

impact of correlated non-Gaussian process variations. Firstly, we need to develop a new set of stochastic basis functions to capture the effects of non-Gaussian correlated process variations. Soize [39] suggested to modify the generalized polynomial chaos, but the resulting non-polynomial basis functions are non-smooth and numerically unstable. Secondly, we need to develop a spectral method (either stochastic collocation or stochastic Galerkin) to compute the weights of the new basis functions. This requires performing a projection step by an accurate numerical integration in a multi-dimensional correlated parameter space. While the numerical integration in a one-dimensional space [40] or a two-dimensional correlated square space [41] is well-studied, accurate numerical integration in a higher-dimensional correlated parameter space remains a challenge. During the preparation of this manuscript, the authors noticed some recent results on stochastic Galerkin [42], [43] and sensitivity analysis for dependent random parameters [44]. However, the theoretical analysis and numerical implementation of stochastic collocation have not been investigated for systems with non-Gaussian correlated parameters.

Main Contributions. This paper presents a novel stochastic collocation approach for systems with correlated non-Gaussian uncertain parameters. Our main contributions include:

- The development of a set of basis functions that can capture the impact of non-Gaussian correlated process variations. Some numerical implementation techniques are also presented to speed up the computation.
- An optimization-based quadrature rule to perform projection in a multi-dimensional correlated parameter space. Previous stochastic spectral methods use id [45] or Gauss quadrature [40], which is not applicable for non-Gaussian correlated cases. We reformulate the numerical quadrature problem as a nonlinear optimization problem, and apply a block coordinate descent method to solve it. Our approach can automatically determinate the number of quadrature samples. We also provide a theoretical analysis for the the upper and lower bounds of the number of quadrature samples required in our framework.
- Theoretical error bound of our algorithm. We show that: (1) we can obtain the exact solution under some mild conditions when the stochastic solution is a polynomial function; (2) for a general smooth stochastic solution, an upper error bound exists for our stochastic collocation algorithm, and it depends on the distance of the unknown solution to a polynomial set as well as the numerical error of our optimization-based quadrature rule.
- A set of numerical experiments on synthetic and realistic electronic and photonic IC examples. The results show the fast convergence rate of our method and its orders-of-magnitude (700× to 6000×) speedup than Monte Carlo.

Before discussing about the technical details, we summarize some of the frequently used notations in Table I.

TABLE I
NOTATIONS IN THIS PAPER

d	number of random parameters describing process variations
p	the highest total polynomial order
M	number of quadrature nodes
$\boldsymbol{\xi}$	a vector denoting d uncertain parameters
$\rho(\boldsymbol{\xi})$	the joint probability density function of $\boldsymbol{\xi}$
$\boldsymbol{\xi}_k$	the value of $\boldsymbol{\xi}$ at a quadrature node
w_k	nonnegative weight associated with $\boldsymbol{\xi}_k$
$\mathbf{w}$	d-dimensional vector of w_k
$\mathbf{e}_1$	a vector of the form $[1, 0, \ldots, 0]^T$
$\boldsymbol{\alpha}$	d-dimensional vector indicating order of a multivariate polynomial
$\Psi_{\boldsymbol{\alpha}}(\boldsymbol{\xi})$	orthonormal basis functions with $\mathbb{E}[\Psi_{\boldsymbol{\alpha}}(\boldsymbol{\xi})\Psi_{\boldsymbol{\beta}}(\boldsymbol{\xi})] = \delta_{\boldsymbol{\alpha}\boldsymbol{\beta}}$
$c_{\boldsymbol{\alpha}}$	coefficient or weight of $\Psi_{\boldsymbol{\alpha}}(\boldsymbol{\xi})$ in the expansion
$\tilde{c}_{\boldsymbol{\alpha}}$	approximation for $c_{\boldsymbol{\alpha}}$ by numerical integration
$\Psi_j(\boldsymbol{\xi})$	$\Psi_{\boldsymbol{\alpha}}(\boldsymbol{\xi})$ in the graded lexicographic order
c_j	coefficient or weight of $\Psi_j(\boldsymbol{\xi})$ in the expansion
$\tilde{c}_j$	approximation for c_j by numerical integration
$y(\boldsymbol{\xi})$	the unknown stochastic solution to be computed
$\tilde{y}(\boldsymbol{\xi})$	approximation of $y(\boldsymbol{\xi})$ by our method
$y_p(\boldsymbol{\xi})$	the projection of $y(\boldsymbol{\xi})$ onto polynomial set $\mathcal{S}_p$
$\mathcal{S}_p$	the set of d-dimensional polynomials with total order $\leq p$
N_p	the number of d-dimensional monomials with order $\leq p$

II. PRELIMINARIES

A. Review of Stochastic Collocation

Stochastic collocation [46]–[49] is the most popular non-intrusive stochastic spectral method. The key idea is to approximate the unknown stochastic solution as a linear combination of some specialized basis functions, and to compute the weights of all basis functions based on a post-processing step such as projection. In order to implement the projection, one needs to do some device- or circuit-level simulations repeatedly for some parameter samples selected by a quadrature rule. Given a good set of basis functions and an accurate quadrature rule, stochastic collocation may obtain a highly accurate result with only a few repeated simulations and can achieve orders-of-magnitude speedup than Monte Carlo when the number of random parameters is small or medium.

Specifically, let $\boldsymbol{\xi} = [\xi_1, \cdots, \xi_d]^T \in \mathbb{R}^d$ denotes a set of random parameters that describe some process variations. We aim to estimate the uncertainty of $y(\boldsymbol{\xi})$, which is a parameter-dependent output of interest such as the power dissipation of a memory cell, the 3-dB band width of an amplifier or the frequency of an oscillator. In almost all chip design cases, we do not have a closed-form expression of $y(\boldsymbol{\xi})$, and we have to call a time-consuming device- or circuit-level simulator (which involves solving large-scale differential equations) to obtain the numerical value of $y(\boldsymbol{\xi})$ for each specified sample of $\boldsymbol{\xi}$. Stochastic spectral methods aim to approximate $y(\boldsymbol{\xi})$ via

$$y(\boldsymbol{\xi}) \approx \sum_{|\boldsymbol{\alpha}|=0}^{p} c_{\boldsymbol{\alpha}} \Psi_{\boldsymbol{\alpha}}(\boldsymbol{\xi}), \text{ with } \mathbb{E}\left[\Psi_{\boldsymbol{\alpha}}(\boldsymbol{\xi})\Psi_{\boldsymbol{\beta}}(\boldsymbol{\xi})\right] = \delta_{\boldsymbol{\alpha},\boldsymbol{\beta}}. \quad (1)$$

Here $\mathbb{E}$ denotes the expectation operator, δ denotes a Delta function, the basis functions $\{\Psi_{\boldsymbol{\alpha}}(\boldsymbol{\xi})\}$ are some orthonormal basis functions indexed by a vector $\boldsymbol{\alpha} = [\alpha_1, \cdots, \alpha_d] \in \mathbb{N}^d$. The total order of the basis function $|\boldsymbol{\alpha}| = \alpha_1 + \ldots + \alpha_d$ is bounded by p, and thus the total number of basis functions is

$$N_p = \binom{p+d}{d} = (p+d)!/(p!d!). \quad (2)$$

The coefficient $c_{\boldsymbol{\alpha}}$ can be obtained by a projection

$$c_{\boldsymbol{\alpha}} = \mathbb{E}\left[y(\boldsymbol{\xi})\Psi_{\boldsymbol{\alpha}}(\boldsymbol{\xi})\right] = \int\limits_{\mathbb{R}^d} y(\boldsymbol{\xi})\Psi_{\boldsymbol{\alpha}}(\boldsymbol{\xi})\rho(\boldsymbol{\xi})d\boldsymbol{\xi}, \quad (3)$$

where $\rho(\boldsymbol{\xi})$ is the joint probability density function. The integral in (3) needs to be evaluated with numerical integration

$$c_{\boldsymbol{\alpha}} \approx \sum_{k=1}^{M} y(\boldsymbol{\xi}_k)\Psi_{\boldsymbol{\alpha}}(\boldsymbol{\xi}_k)w_k. \tag{4}$$

where $\{\boldsymbol{\xi}_k\}_{k=1}^{M}$ are the quadrature nodes, and $\{w_k\}_{k=1}^{M}$ are the corresponding quadrature weights. The key of stochastic collocation is to choose proper basis functions and an excellent quadrature rule, such that M is as small as possible in (4).

B. Existing Solutions for Independent Cases

Most existing stochastic spectral methods assume that $\boldsymbol{\xi} = [\xi_1, \ldots, \xi_d]^T$ are mutually independent. In this case, given the marginal density function $\rho_k(\xi_k)$ of each parameter, the joint density function is $\rho(\boldsymbol{\xi}) = \Pi_{k=1}^{d}\rho_k(\xi_k)$. Consequently, an excellent choice of basis functions is the generalized polynomial chaos [10]: the multivariate basis function is obtained as the product of some univariate polynomial basis functions

$$\Psi_{\boldsymbol{\alpha}}(\boldsymbol{\xi}) = \phi_{1,\alpha_1}(\xi_1)\ldots\phi_{d,\alpha_d}(\xi_d). \tag{5}$$

Here each univariate basis function $\phi_{k,\alpha_k}(\xi_k)$ can be constructed via the well-known three-term recurrence relation [50], and the univariate basis functions of the same parameter ξ_k are mutually orthonormal with respect to the marginal density function $\rho_k(\xi_k)$.

When $\boldsymbol{\xi}$ are mutually independent, the quadrature points and weights in (4) are often constructed via the tensor product of one-dimensional quadrature points and weights. Specifically, denote $\{\xi_{i_k}, w_{i_k}\}$ as the quadrature nodes and weights for a parameter ξ_k (for instance, via Gaussian quadrature rule [40]), then $\boldsymbol{\xi}_{i_1\ldots i_d} = [\xi_{i_1}, \ldots, \xi_{i_d}]^T$ and $w_{i_1\ldots i_d} = w_{i_1}\ldots w_{i_d}$ are the quadrature points and weights for a d-dimensional problem. Another popular approach is the sparse grid technique [45], [51]–[53], which can significantly reduce the number of quadrature points by exploiting the nested structure of the quadrature points of different accuracy levels.

C. Non-Gaussian Correlated Cases

In general, $\boldsymbol{\xi}$ can be non-Gaussian correlated, and the joint density $\rho(\boldsymbol{\xi})$ cannot be written as the product of the individual marginal density functions. As a result the multivariate basis function can not be obtained as in (5). It is also hard to choose a small number of quadrature nodes $\{\boldsymbol{\xi}_k\}$ and weights $\{w_k\}$ that can produce highly accurate integration results.

In order to quantify the impact of non-Gaussian correlated uncertainties, Soize [39] suggested a set of non-smooth orthonormal basis functions by modifying the generalized polynomial chaos [10]. The modified basis functions were employed in [26] for the variability analysis of silicon photonic devices. However, the algorithm does not converge well due to the numerical instability of the basis functions, and designers cannot easily extract statistical information (e.g., mean value and variance) from the obtained solution. In the applied math community, multivariate orthogonal polynomials may be constructed via the multivariate three-term recurrence [54], [55]. However, the theories in [54], [55] either are hard to implement or can only guarantee weak orthogonality.

III. Proposed Orthonormal Basis Functions

This section presents a set of smooth orthonormal basis functions that can capture the impact of non-Gaussian correlated random parameters. The proposed basis functions allow us to approximate a smooth $y(\boldsymbol{\xi})$ with a high accuracy and to extract its statistical moments analytically or semi-analytically.

A. Generating Multivariate Orthonormal Polynomials

We adopt a Gram-Schmidt approach to calculate the basis functions recursively. The Gram-Schmidt method was used for vector orthogonalization in the Euclidean space [56]. It can also be generalized to construct some orthogonal polynomial functions. The key difference here is to replace the vector inner product with the functional expectations.

Specifically, we first reorder the monomials $\boldsymbol{\xi}^{\boldsymbol{\alpha}} = \xi_1^{\alpha_1}\ldots\xi_d^{\alpha_d}$ in the graded lexicographic order, and denote them as $\{p_j(\boldsymbol{\xi})\}_{j=1}^{N_p}$. For instance, when $d=2$ and $p=2$, there is

$$\{p_j(\xi_1,\xi_2)\}_{j=1}^{6} = \{1, \xi_1, \xi_2, \xi_1^2, \xi_1\xi_2, \xi_2^2\}.$$

Then we set $\Psi_1(\boldsymbol{\xi}) = 1$ and generate orthonormal polynomials $\{\Psi_j(\boldsymbol{\xi})\}_{j=1}^{N_p}$ in the correlated parameter space recursively by

$$\hat{\Psi}_j(\boldsymbol{\xi}) = p_j(\boldsymbol{\xi}) - \sum_{i=1}^{j-1}\mathbb{E}[p_j(\boldsymbol{\xi})\Psi_i(\boldsymbol{\xi})]\Psi_i(\boldsymbol{\xi}), \tag{6}$$

$$\Psi_j(\boldsymbol{\xi}) = \frac{\hat{\Psi}_j(\boldsymbol{\xi})}{\sqrt{\mathbb{E}[\hat{\Psi}_j^2(\boldsymbol{\xi})]}},\ j = 2, \ldots, N_p. \tag{7}$$

The basis functions defined by this approach are unique under the specific order of monomials. If the ordering of monomials are changed, one can get another set of basis functions. Since the basis functions are orthonormal polynomials, we can easily extract the mean value and statistical moment of an approximated stochastic solution.

Recently we have also proposed basis function construction method via a Cholesky decomposition [57], which is easy to implement and suitable for high-dimensional cases. However, the resulting basis functions can be occasionally inaccurate due to the numerical instability of the Cholesky factorization on large ill-conditioned covariance matrices. This paper focuses on the fundamental theory of stochastic collocation for correlated cases, therefore, we employ the Gram-Schmidt method.

B. Numerical Implementation Issues

The main challenge in the basis function generation is to compute the expectations in a correlated parameter space, which involves evaluating the moments $\mathbb{E}[\boldsymbol{\xi}^{\boldsymbol{\alpha}}]$ up to order $2p$. Some techniques can be used to speed up the computation.

In practice, the process variations are generally described by a set of measurement data from testing chips, and their joint density function $\rho(\boldsymbol{\xi})$ is fitted using some density estimators. A widely used model is the Gaussian mixture:

$$\rho(\boldsymbol{\xi}) = \sum_{i=1}^{n} r_i\mathcal{N}(\boldsymbol{\xi}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i),\ \text{with } r_i > 0,\ \sum_{i=1}^{n} r_i = 1. \tag{8}$$

Here $\mathcal{N}(\boldsymbol{\xi}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$ denotes a multi-variate Gaussian distribution with mean $\boldsymbol{\mu}_i \in \mathbb{R}^d$ and a covariance matrix $\boldsymbol{\Sigma}_i \in \mathbb{R}^{d\times d}$.

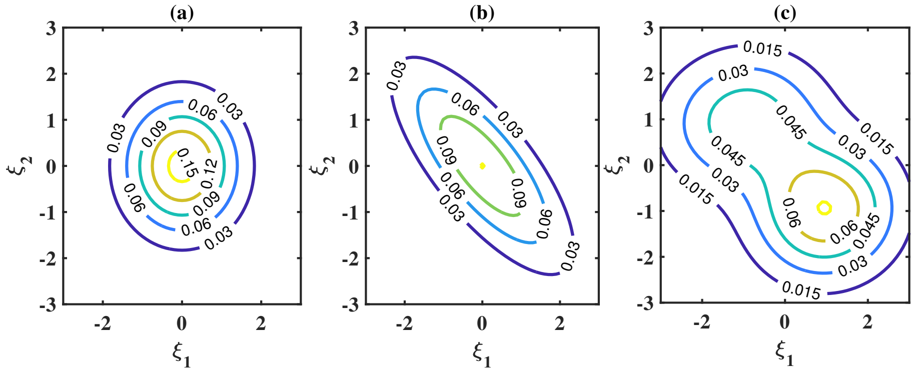


Fig. 1. Several joint density functions. (a) independent Gaussian; (b) correlated Gaussian; (c) correlated non-Gaussian (e.g., a Gaussian-mixture distribution).

Fig. 1 compares the Gaussian mixture model with independent and correlated Gaussian distributions. With a Gaussian mixture, the moments can be computed accurately using a functional tensor train approach (see Section 3.3 of [57]).

For general cases, one may estimate the moments by changing the variables and density function:

$$\mathbb{E}[\boldsymbol{\xi}^{\boldsymbol{\alpha}}] = \int_{\mathbb{R}^d} g_{\boldsymbol{\alpha}}(\boldsymbol{\eta})\hat{\rho}(\boldsymbol{\eta})d\boldsymbol{\eta}, \text{ with } g_{\boldsymbol{\alpha}}(\boldsymbol{\eta}) = \frac{\boldsymbol{\eta}^{\boldsymbol{\alpha}}\rho(\boldsymbol{\eta})}{\hat{\rho}(\boldsymbol{\eta})}. \tag{9}$$

Here $\hat{\rho}(\boldsymbol{\eta})$ denotes the joint density function of independent random parameters $\boldsymbol{\eta} \in \mathbb{R}^d$. Then, standard quadrature methods such as sparse grid [45] or tensor-product Gauss quadrature can be used to evaluate the integration. The tensor-train-based method in [25] can be used to reduce the integration cost when d is large. The potential limitation is that it may be non-trivial to obtain highly accurate results if $g_{\boldsymbol{\alpha}}(\boldsymbol{\eta})$ is highly nonlinear or even non-smooth. Note that we only need to use a high-order quadrature rule in an independent parameter space and repeatedly evaluate some cheap closed-form functions here, and we do not need to perform expensive device or circuit simulations when we compute the basis functions.

In this paper, we use Gaussian mixture models to describe non-Gaussian correlated uncertainties, and we employ the functional tensor-train method [57] for moment computation.

IV. Optimization-Based Quadrature

After constructing the basis functions, we still need to choose a small number of the quadrature nodes and weights in order to calculate $c_{\boldsymbol{\alpha}}$ by (4) with a small number of device- or circuit-level simulations. Motivated by [58], [59], we present an optimization model to decide a proper quadrature rule. Our method differs from [58], [59] in both algorithm framework and theoretical analysis. Firstly, while [58] only updates the quadrature weights by linear programing, we optimize the quadrature samples and weights by nonlinear optimization. Secondly, our optimization setup differs from that in [59]: we minimize the integration error of our proposed multivariate orthonormal basis functions, such that the resulting quadrature rule is suitable for quantifying the impact of non-Gaussian correlated uncertainties. Thirdly, we handle the nonnegative constraint of the weight $\mathbf{w}$ and the nonlinear objective function of $\bar{\boldsymbol{\xi}}$ separately via a block coordinate descent approach. Fourthly, we propose a novel initializing method via weighted complete linkage clustering. Finally, we present theoretical results regarding the algorithm complexity and error bound. Our method is summarized in Algorithm 1, and we elaborate the key ideas below.

```
Algorithm 1: Proposed stochastic collocation method
Step 1  Initialize the quadrature nodes and weights via
        Algorithm 3.
Step 2  Increase phase. Update the quadrature nodes and
        weights by solving (11). If Alg. 2 fails to converge,
        increase the node number and go back to Step 1.
Step 3  Decrease phase. Decrease the number of nodes, and
        update them by solving (11). Repeat Step 3 until no
        points can be deleted [in other words, the objective
        function of (11) fails to reduce below a prescribed
        threshold]. Return the nodes and weights.
Step 4  Call a deterministic simulator to compute $\{y(\boldsymbol{\xi}_k)\}_{k=1}^{M}$.
        Then compute the coefficients $\{c_{\boldsymbol{\alpha}}\}$ via (4).
  Output: The coefficients $\{c_{\boldsymbol{\alpha}}\}$ in (1).
```

A. Optimization Model of Our Quadrature Rule

Our idea is to compute a set of quadrature points and weights that can accurately estimate the numerical integration of some testing functions. Given a joint density function $\rho(\boldsymbol{\xi})$, we seek for the quadrature nodes and weights $\{\boldsymbol{\xi}_k, w_k\}_{k=1}^{M}$ by matching the integration of basis functions up to order $2p$:

$$\mathbb{E}[\Psi_j(\boldsymbol{\xi})] = \int_{\mathbb{R}^d} \Psi_j(\boldsymbol{\xi})\rho(\boldsymbol{\xi})d\boldsymbol{\xi} = \sum_{k=1}^{M} \Psi_j(\boldsymbol{\xi}_k)w_k, \qquad \forall j = 1, \ldots, N_{2p}. \tag{10}$$

Here, $N_{2p} = \binom{2p+d}{d}$ denotes the total number of basis functions with their total order bounded by $2p$.

We choose the above testing functions based on two reasons. Firstly, it is easy to show that $\mathbb{E}[\Psi_j(\boldsymbol{\xi})] = \mathbb{E}[\Psi_j(\boldsymbol{\xi})\Psi_1(\boldsymbol{\xi})] = \delta_{1j}$. Secondly, we can show that for any polynomial function $f(\boldsymbol{\xi})$ bounded by order $2p$, the integration of $f(\boldsymbol{\xi})$ weighted by the density function $\rho(\boldsymbol{\xi})$ (i.e., $\mathbb{E}\left[f(\boldsymbol{\xi})\right]$) can be written as the weighted sum of $\mathbb{E}[\Psi_j(\boldsymbol{\xi})]$'s, and therefore one can get the exact integration result if (10) holds. In stochastic collocation, if $y(\boldsymbol{\xi})$ is a polynomial function bounded by order p, then $c_{\boldsymbol{\alpha}} = \mathbb{E}\left[y(\boldsymbol{\xi})\Psi_{\boldsymbol{\alpha}}(\boldsymbol{\xi})\right]$ can be accurately computed for *every* basis function with $|\boldsymbol{\alpha}| \leq p$ if (10) holds. The detailed derivations are given in Theorem 2 of Section V.

In practice, we propose to rewrite (10) as the following nonlinear least-square problem

$$\min_{\bar{\boldsymbol{\xi}},\mathbf{w}\geq 0} \quad \|\boldsymbol{\Phi}(\bar{\boldsymbol{\xi}})\mathbf{w} - \mathbf{e_1}\|_2^2, \tag{11}$$

where $\bar{\boldsymbol{\xi}} = [\boldsymbol{\xi}_1^T, \ldots, \boldsymbol{\xi}_M^T]^T \in \mathbb{R}^{Md}$, $\mathbf{w} = [w_1, \ldots, w_M]^T \in \mathbb{R}^M$, $\mathbf{e}_1 = [1, 0, \ldots, 0]^T \in \mathbb{R}^{N_{2p}}$, $\boldsymbol{\Phi}(\bar{\boldsymbol{\xi}})$ is a matrix of size $N_{2p} \times M$ with the (j,k)-th element being $(\boldsymbol{\Phi}(\bar{\boldsymbol{\xi}}))_{jk} = \Psi_j(\boldsymbol{\xi}_k)$, $\|\cdot\|_2$ denotes the Euclidean norm. Here, we also require the quadrature weights to be nonnegative. This requirement is an natural extension of the one-dimensional Gauss quadrature rule [40], and it can help our theoretical analysis in Section V.

B. A Block Coordinate Descent Solver for (11)

The total number of unknowns in (11) is $M(d+1)$, which becomes large as d increases. In order to improve the scalability of our algorithm, we solve (11) by a block coordinate descent method. The idea is to update the parameters block-by-block: at the t-th iteration, we firstly fix $\bar{\boldsymbol{\xi}}^{t-1}$ and solve a $\mathbf{w}$-subproblem to update $\mathbf{w}^t$, then fix $\mathbf{w}^t$ and solve a $\boldsymbol{\xi}$-subproblem to update $\bar{\boldsymbol{\xi}}^t$.

w-subproblem. Suppose $\bar{\boldsymbol{\xi}}^{t-1} = [\boldsymbol{\xi}_1^{t-1}; \ldots; \boldsymbol{\xi}_M^{t-1}]$ is fixed, then (11) reduces to a convex linear least-square problem

$$\mathbf{w}^t = \arg\min_{\mathbf{w}\geq 0} \quad \|\boldsymbol{\Phi}(\bar{\boldsymbol{\xi}}^{t-1})\mathbf{w} - \mathbf{e}_1\|_2^2. \tag{12}$$

$\boldsymbol{\xi}$-subproblem. When $\mathbf{w}^t$ is fixed, we apply the Gaussian Newton method to update the quadrature samples:

$$\boldsymbol{\xi}_k^t = \boldsymbol{\xi}_k^{t-1} + \mathbf{d}_k^t, \text{ with } \{\mathbf{d}_k^t\} = \arg\min_{\{\mathbf{d}_k\}} \; \|\sum_{k=1}^{M} \mathbf{G}_k^t \mathbf{d}_k + \mathbf{r}^t\|_2^2. \tag{13}$$

Here, $\mathbf{r}^t = \boldsymbol{\Phi}(\bar{\boldsymbol{\xi}}^{t-1})\mathbf{w}^t - \mathbf{e}_1 \in \mathbb{R}^{N_{2p}}$ denotes the residual, $\mathbf{G}_k^t \in \mathbb{R}^{N_{2p}\times d}$ is the Jacobian matrix of $\mathbf{r}^t$ with respect to $\boldsymbol{\xi}_k^{t-1}$. In practice, we run the step in (13) once and go back to the $\mathbf{w}$-step. This is actually the inexact block coordinate approach [60]. The pseudo codes of our block coordinate descent solver are summarized in Algorithm 2. Here we use an ℓ_1-norm in the stopping criteria since it enables us to bound the error of our whole framework in Section V.

We note that some other approaches can also solve the non-convex optimization problem (11). When the number of unknown variables is small, we can obtain a globally optimal solution via the polynomial optimization solver based on a semi-definite positive relaxation [61]. The Levenberg-Marquardt approach or the trust region algorithm [62] can also be used to solve the $\boldsymbol{\xi}$-subproblem, but they are more expensive than our solver. Our optimization solver converges very well in practice. As will be shown in Section V, our stochastic collocation framework actually does not necessarily require a locally or globally optimal solution of (11) at all. Instead, it only requires the objective function to be sufficiently small at the obtained quadrature samples and weights.

Algorithm 2: Block coordinate descent solver for (11)

```
Input: Initial quadrature nodes $\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_M$, the maximal
       iteration $n_{\max}$, and the tolerance $\epsilon$.
for $t = 1, \ldots, n_{\max}$ do
    Update the weights $\mathbf{w}^t$ via solving (12);
    Update the nodes $\bar{\boldsymbol{\xi}}^t$ via solving (13);
    if $\|\boldsymbol{\Phi}(\bar{\boldsymbol{\xi}}^t)\mathbf{w}^t - \mathbf{e_1}\|_1 \leq \epsilon$ is satisfied then
        break;
Output: Optimal nodes and weights $\{\boldsymbol{\xi}_k, w_k\}_{k=1}^M$.
```

C. Initializing Quadrature Nodes and Weights

The nonlinear least square problem (11) is non-convex, and generally it is hard to obtain the global optimal solution. In practice, accurate results can be obtained once we can use good initial guesses for the quadrature nodes and weights.

In Step 3 of Algorithm 1, we need to find a quadrature rule with fewer nodes after some pairs of quadrature samples and weights have already been calculated. In this case, we can simply delete one node with the smallest weight, and choose all other samples and their corresponding weights as the initial condition for the subsequent optimization problem.

In Step 1 of Algorithm 1, we need to generate some initial nodes from scratch. We firstly generate $M_0 \gg M$ nodes via Monte Carlo. In Monte Carlo sampling, all samples have the same weights $1/M_0$. In order to improve the convergence, we keep all samples unchanged but refine their weights by solving the w-subproblem in (12). These M_0 initial nodes are then grouped into M clusters, and the resulting cluster centers are set as the initial samples for whole nonlinear least-square optimization problem. This choice of initial guess proves to work very well in practice, because Monte Carlo itself is an integration rule with statistical accuracy guarantees.

Clustering is a classical technique in pattern recognition and data mining [63], and it gathers data with similar pattern into one group. A widely used algorithm is hierarchical clustering. At the beginning, each single data point is a cluster by its own, then two clusters with "the minimal distance" are merged into one single cluster sequentially. Consequently, the number of clusters is decreased by one in each iteration until the prescribed number of clusters is reached. The widely used hierarchical approaches includes single linkage, complete linkage and average linkage. They mainly differ in the criterion of choosing "the distance". The complete-linkage clustering chooses the distance between two clusters C_i and C_j as

$$D_{ij}^0 = \max_{\boldsymbol{\xi}_1\in C_i, \boldsymbol{\xi}_2\in C_j} d(\boldsymbol{\xi}_1, \boldsymbol{\xi}_2),$$

where $d(\boldsymbol{\xi}_1, \boldsymbol{\xi}_2) = \|\boldsymbol{\xi}_1 - \boldsymbol{\xi}_2\|_2$. In our problem, the sample points are equipped with some weight parameters, therefore, we modify the complete-linkage clustering and consider a weighted clustering problem.

Weighted Complete Linkage Clustering. We define the weighted distance as

$$D_{ij} = (w_i + w_j)\left(\max_{\boldsymbol{\xi}_1\in C_i, \boldsymbol{\xi}_2\in C_j} d(\boldsymbol{\xi}_1, \boldsymbol{\xi}_j)\right), \tag{14}$$

```
Algorithm 3: Weighted complete linkage clustering
  Input: The number of cluster $M$, and $M_0 = 3M$ initial
         nodes $\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_{M_0}$.
  Calculate the weights for $\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_{M_0}$ by solving (12).
  for $m = M_0, \ldots, M+1$ do
      Update the distance matrix by (14).
      Find two clusters with the minimal distance, and
        merge them into one single cluster.
  Calculate the cluster centers and weights via (15).
  Output: Clustered nodes and weights $\{\boldsymbol{\xi}_k, w_k\}_{k=1}^{M}$.
```

where $w_i = \sum_{\boldsymbol{\xi}_k \in C_i} w(\boldsymbol{\xi}_k)$ is the weight of the i-th cluster. The above distance considers both the geometric distance and the weights of different clusters. The intuition behind (14) is that we do not want a sample with a very small weight to form a cluster by itself. This algorithm tends to group a sample with a very small weight with its nearest cluster.

Once the number of clusters reduces to M, we stop the iterations and return the weight and cluster center as

$$w_i = \sum_{\boldsymbol{\xi}_k \in \mathbf{C}_i} w(\boldsymbol{\xi}_k),\ \boldsymbol{\xi}_i = \sum_{\boldsymbol{\xi}_k \in \mathbf{C}_i} \frac{w(\boldsymbol{\xi}_k)}{w_i}\boldsymbol{\xi}_k,\ \forall\, i = 1, \ldots, M. \tag{15}$$

Algorithm 3 has summarized the pseudo codes of our clustering method used to initialize Algorithm 1.

D. Number of Quadrature Points

A fundamental question is: how many quadrature samples are necessary in order to achieve a desired level of accuracy? This question is well answered in the one-dimensional Gauss quadrature rule: p quadrature points provide an exact result for the numerical integration of any polynomial function bounded by order $2p-1$ [40]. However, there is no similar result for general multidimensional correlated cases.

Let $\mathcal{S}_{2p}$ denote all polynomial functions of $\boldsymbol{\xi}$ with their total orders bounded by $2p$. The integration rule $\{\boldsymbol{\xi}_k, w_k\}_{k=1}^{M}$ has a $2p$-th-order accuracy if (10) is satisfied. Here the $2p$-th-order accuracy means that $\sum\limits_{k=1}^{M} f(\boldsymbol{\xi}_k)w_k = \mathbb{E}[f(\boldsymbol{\xi})]$ for *any* $f(\boldsymbol{\xi}) \in \mathcal{S}_{2p}$. We have the following result on the number of quadrature samples in order to ensure the $2p$-th-order accuracy.

Theorem 1. *Assume that M pairs of quadrature samples and weights are obtained from* (10) *to ensure the $2p$-th-order integration accuracy, then the number of quadrature points satisfies $N_p \leq M \leq N_{2p}$.*

Proof. See Appendix A for the details. □

While there exists at least one M in $[N_p, N_{2p}]$ such that the $2p$-th-order integration accuracy can be achieved, we can have multiple choices of M, and we may even have multiple choices of quadrature samples and weights for each M. In our stochastic collocation framework, we only require one (among possibly multiple) set of quadrature samples and weights with a sufficiently small M.

In practice, we try to get a better solution by generating a better initial guess. We do this by firstly generate $6N_p$ random samples via Monte Carlo, and group them into $2N_p$ clusters. These $M = 2N_p$ samples are used as the initial quadrature points. Then, we increase or decrease M via Algorithm 1. This process is illustrated via a 2-dimensional example in Fig 2. The practical number of quadrature nodes used by our stochastic collocation framework is very close to the theoretical lower bound, which is experimentally shown in Section VI-E.

V. Theoretical Error Bounds

In this section, we provide several theoretical results regarding the numerical accuracy of our proposed stochastic collocation algorithm for non-Gaussian correlated cases.

A. Conditions for Exact Results

The following theorem show that our quadrature rule (10) can provide exact results if $y(\boldsymbol{\xi})$ satisfies certain conditions.

Theorem 2. *Suppose that $y(\boldsymbol{\xi}) \in \mathcal{S}_p$ is a polynomial function bounded by order p, i.e., there exist some coefficients $\{c_{\boldsymbol{\alpha}}\}$ such that $y(\boldsymbol{\xi}) = \sum_{|\boldsymbol{\alpha}|=0}^{p} c_{\boldsymbol{\alpha}}\Psi_{\boldsymbol{\alpha}}(\boldsymbol{\xi})$. Denote the approximated expansion obtained via our numerical integration as*

$$\tilde{y}(\boldsymbol{\xi}) = \sum_{|\boldsymbol{\alpha}|=0}^{p} \tilde{c}_{\boldsymbol{\alpha}}\Psi_{\boldsymbol{\alpha}}(\boldsymbol{\xi}),\ \textit{with}\ \tilde{c}_{\boldsymbol{\alpha}} = \sum_{k=1}^{M} y(\boldsymbol{\xi}_k)\Psi_{\boldsymbol{\alpha}}(\boldsymbol{\xi}_k)w_k. \tag{16}$$

Then $y(\boldsymbol{\xi})$ can be recovered exactly, i.e., $y(\boldsymbol{\xi}) = \tilde{y}(\boldsymbol{\xi})$, if $\{\boldsymbol{\xi}_k, w_k\}$ satisfies (10) *strictly for all $j = 1, \ldots, N_{2p}$.*

Proof. The detailed proof is provided in Appendix B. □

In practice, we may not be able to get an exact solution because of two reasons: (1) $y(\boldsymbol{\xi})$ is not a polynomial in $\mathcal{S}_p$; (2) the quadrature points and weights obtained by our numerical nonlinear optimization solver causes a small residual in (10). In this case, we can provide an error bound for our solution when $y(\boldsymbol{\xi})$ is smooth enough and when the nonlinear optimization problem (11) is solved with certain accuracy (i.e., when the resulting objective function is below a threshold).

B. Three Weak Assumptions

In order to provide a theoretical analysis for the numerical error caused by $y(\boldsymbol{\xi})$ and by the nonlinear optimization solver, we make the following weak assumptions.

Assumption 1. $y(\boldsymbol{\xi})$ is squared integrable. In other words, there exists a positive scalar L such that

$$\|y(\boldsymbol{\xi})\|_2 = \sqrt{\mathbb{E}[y^2(\boldsymbol{\xi})]} \leq L. \tag{17}$$

Let $y_p(\boldsymbol{\xi}) = \arg\min_{\hat{y}(\boldsymbol{\xi}) \in \boldsymbol{\mathcal{S}}_p} \|y(\boldsymbol{\xi}) - \hat{y}(\boldsymbol{\xi})\|_2$ be the projection of $y(\boldsymbol{\xi})$ onto $\boldsymbol{\mathcal{S}}_p$, and assume that there exists δ such that

$$0 \leq \|y(\boldsymbol{\xi}) - y_p(\boldsymbol{\xi})\|_2 \leq \delta. \tag{18}$$

Actually $y_p(\boldsymbol{\xi})$ can be written as $y_p(\boldsymbol{\xi}) = \sum_{|\boldsymbol{\alpha}|=0}^{p} c_{\boldsymbol{\alpha}}\Psi_{\boldsymbol{\alpha}}(\boldsymbol{\xi})$, where $c_{\boldsymbol{\alpha}} = \mathbb{E}[y(\boldsymbol{\xi})\Psi_{\boldsymbol{\alpha}}(\boldsymbol{\xi})]$.

Assumption 2. Define the numerical integration operator

$$\mathbb{I}[y(\boldsymbol{\xi})] = \sum_{k=1}^{M} y(\boldsymbol{\xi}_k)w_k. \tag{19}$$

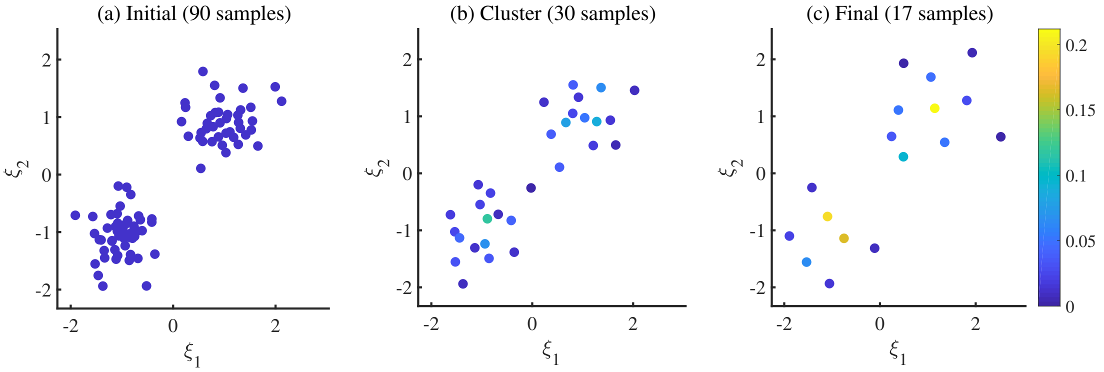


Fig. 2. The process of generating quadrature samples and weights for the synthetic example. The quadrature weights are shown by the color bar. (a) Initial candidate points generated via Monte Carlo; (b) clustered samples via the weighted complete linkage method in Algorithm 3; (c) the optimized quadrature nodes by our Algorithm 1. This process only depends on the probability density function and the basis functions, and is independent of $y(\boldsymbol{\xi})$.

We assume that the operator $\mathbb{I}[y(\boldsymbol{\xi})]$ is bounded, i.e., there exists $W \geq 0$ such that

$$|\mathbb{I}[y(\boldsymbol{\xi})]| \leq W\|y(\boldsymbol{\xi})\|_1, \text{ where } \|y(\boldsymbol{\xi})\|_1 = \mathbb{E}[|y(\boldsymbol{\xi})|]. \quad (20)$$

Assumption 3. The nonlinear least square problem (11) is solved with an error threshold $\epsilon \geq 0$, i.e.,

$$\|\boldsymbol{\Phi}(\bar{\boldsymbol{\xi}})\mathbf{w} - \mathbf{e_1}\|_1 \leq \epsilon, \quad (21)$$

where $\|\cdot\|_1$ denotes the ℓ_1 norm in the Euclidean space. Here the j-th element in the vector $\boldsymbol{\Phi}(\bar{\boldsymbol{\xi}})\mathbf{w} - \mathbf{e_1}$ actually can be written as $\mathbb{I}[\Psi_j(\boldsymbol{\xi})] - \mathbb{E}[\Psi_j(\boldsymbol{\xi})]$.

C. Error Bound of the Proposed Stochastic Collocation

Theorem 3. *Suppose that Assumptions 1-3 hold, then numerical integration error satisfies*

$$|\mathbb{E}[y(\boldsymbol{\xi})] - \mathbb{I}[y(\boldsymbol{\xi})]| \leq L\epsilon + W\delta. \quad (22)$$

Here L is the upper bound of $\|y(\boldsymbol{\xi})\|_2$ in (17)*, W is the upper bound of the numerical integration $\mathbb{I}[y(\boldsymbol{\xi})]$ in* (20)*, ϵ is the numerical error of our nonlinear optimization solver defined in* (21)*, and δ is the distance from $y(\boldsymbol{\xi})$ to $\boldsymbol{\mathcal{S}}_p$ in* (18)*.*

Proof. See Appendix C. □

Based on Theorem 3, we can further derive an upper bound for the following approximation error.

Theorem 4. *With Assumptions 1-3, the numerical error of our stochastic collocation algorithm satisfies*

$$\|y(\boldsymbol{\xi}) - \tilde{y}(\boldsymbol{\xi})\|_2 \leq \delta + N_p(LT\epsilon + W\delta), \quad (23)$$

where $T = \max_{j,l=1,\ldots,N_{2p}} \|\Psi_j(\boldsymbol{\xi})\Psi_l(\boldsymbol{\xi})\|_2$.

Proof. See Appendix D for the details. □

Remarks: Theorem 4 indicates the following intuitions:

1) if the nonlinear optimization solver is accurate enough and ϵ is very small, the error of our stochastic collocation is dominated by the approximation error δ;
2) as we increase the order of basis functions, δ decreases and the result becomes more and more accurate;
3) if the total order of the basis function is very high and δ becomes extremely small, the optimization error ϵ will dominate the overall numerical error, and the convergence will slow down.

Once (10) holds, we should have the following result

$$\mathbb{I}[\Psi_i(\boldsymbol{\xi})\Psi_j(\boldsymbol{\xi})] = \sum_{k=1}^{M} \Psi_i(\boldsymbol{\xi}_k)\Psi_j(\boldsymbol{\xi}_k)w_k = \delta_{i,j}.$$

In practice, there are numerical errors caused by quadrature points and weights obtained by the optimization solver. In the following lemma, we show that the error is bounded.

Lemma 1. *Suppose that Assumptions 1-3 hold, define a matrix $\mathbf{V} \in \mathbb{R}^{N_p \times N_p}$ with each element $\mathbf{V}_{ij} = \mathbb{I}[\Psi_i(\boldsymbol{\xi})\Psi_j(\boldsymbol{\xi})]$ being a numerical evaluation of $\mathbb{E}[\Psi_i(\boldsymbol{\xi})\Psi_j(\boldsymbol{\xi})]$ using the quadrature points and weights from solving* (11)*. We have*

$$\|\mathbf{V} - I_{N_p}\|_F \leq N_pT\epsilon. \quad (24)$$

Proof. See Appendix E. □

VI. Numerical Results

In order to show the efficiency of our proposed method, we conduct numerical experiments on a synthetic example, a 3-stage CMOS electronic ring oscillator, and an optical filter. The stopping criterion in (11) is set as $\epsilon = 10^{-8}$ unless stated otherwise. In all examples, we use some Gaussian mixture models to describe the joint density functions of correlated non-Gaussian random parameters. The Matlab codes and a demo example are provided online at: https://web.ece.ucsb.edu/~zhengzhang/codes_dataFiles/uq_ng.

A. A Synthetic Example

Firstly we consider a synthetic example, and use it to show the accuracy and convergence rate of our proposed stochastic collocation algorithm. Specifically, we consider the following smooth function of two correlated parameters

$$y(\boldsymbol{\xi}) = \exp(\xi_1) + 0.1\cos(\xi_1)\sin(\xi_2). \quad (25)$$

We assume that the random parameters follow a Gaussian mixture distribution

$$\boldsymbol{\xi} = \boldsymbol{\xi}_0 + \frac{1}{10}\Delta\boldsymbol{\xi}, \text{ where } \Delta\boldsymbol{\xi} \sim \frac{1}{2}\mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) + \frac{1}{2}\mathcal{N}(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2).$$

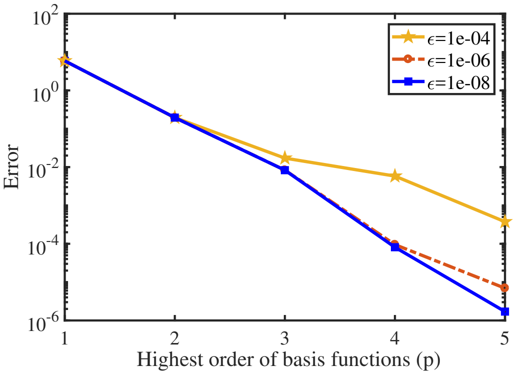


Fig. 3. Convergence rate for the synthetic example. Here ϵ is the numerical error of optimization defined in (21). This figure demonstrates the error estimated in (23): the stochastic collocation algorithm shows a nearly exponential convergence rate as p increases and before ϵ dominates the error.

TABLE II
ACCURACY COMPARISON ON THE SYNTHETIC EXPERIMENTS. THE UNDERSCORES INDICATE PRECISION.

method	Proposed				
p	1	2	3	4	5
# samples	3	6	10	17	66
mean	2.7$\underline{8}$35	2.782$\underline{9}$	2.782$\underline{9}$	2.782$\underline{9}$	2.782$\underline{9}$
method	Monte Carlo				
# samples	10	10^2	10^3	10^4	10^5
mean	$\underline{2}$.6799	2.$\underline{7}$625	2.$\underline{7}$911	2.7$\underline{8}$11	2.782$\underline{9}$

Here, the mean values $\boldsymbol{\mu}_1 = \mathbf{1}$, $\boldsymbol{\mu}_2 = -\mathbf{1}$; the positive definite covariance matrices $\boldsymbol{\Sigma}_1$ and $\boldsymbol{\Sigma}_2$ are randomly generated. We use $\mathbf{1}$ to denote a vector of a compatible size with all elements being one. We will also use this notation in other examples.

We first illustrate how to generate the quadrature samples and weights by our optimization-based quadrature rule. Assume that we want to approximate $y(\boldsymbol{\xi})$ by a forth-order expansion. Firstly, 90 random samples are generated via Monte Carlo. Secondly, these points are grouped into 30 clusters via our proposed weighted linkage clustering approach, and they are used as the initial samples and weights of Algorithm 1. Finally, the number of quadrature nodes is reduced to 17 automatically by Algorithm 1, whereas the lower bound for the number of quadrature nodes is 15. The process of generating quadrature samples and weights is shown in Fig. 2.

Theorem 4 shows that the error depends on two parts: the numerical error ϵ of the optimization solver of our quadrature rule, and the approximation error δ by order-p basis functions. When p is small, $\|y(\boldsymbol{\xi}) - y_p(\boldsymbol{\xi})\|_2 \leq \delta$ dominates the error. When p is large, δ becomes small and ϵ dominates the error, therefore smaller ϵ will produce more accurate results. In order to verify this theoretical result, we perform stochastic collocation by using different orders of basis functions (i.e., $p = 1$ to 5) and by setting different error thresholds (i.e., $\epsilon = 10^{-4}$, 10^{-6} and 10^{-8}) in the optimization-based quadrature rule. As shown in Fig. 3, our stochastic collocation has a nearly exponential convergence rate before ϵ dominates the error.

We further compare our our method with Monte Carlo in Table II. Our method provides a closed-form expression for the mean value of $y(\boldsymbol{\xi})$, and a 2nd-order expansion using 6 quadrature points is sufficient to achieve a precision of 4 fractional digits. In contrast, Monte Carlo requires 10^5 random samples to achieve the similar level of accuracy.

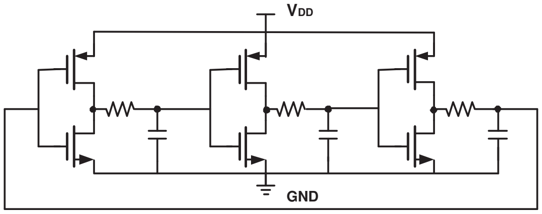


Fig. 4. Schematic of a 3-stage CMOS ring oscillator.

TABLE III
ACCURACY COMPARISON ON THE CMOS RING OSCILLATOR. THE UNDERSCORES INDICATE PRECISION.

method	Proposed	Monte Carlo			
# samples	33	10^2	10^3	10^4	10^5
mean (MHz)	30.8$\underline{3}$	3$\underline{0}$.93	30.$\underline{8}$8	30.$\underline{8}$0	30.8$\underline{3}$

B. A 3-Stage CMOS Electronic Ring Oscillator

We continue to verify our algorithm by the 3-stage CMOS ring oscillator in Fig. 4. We model the relative threshold voltage variations of six transistors via

$$\boldsymbol{\xi} = \boldsymbol{\xi}_0 + \mathbf{D}\Delta\boldsymbol{\xi}, \text{ with } \Delta\boldsymbol{\xi} \sim \frac{2}{3}\mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) + \frac{1}{3}\mathcal{N}(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2),$$

where $\mathbf{D}$ is a diagonal scaling matrix, $\mu_1 = \mathbf{1}$, $\mu_2 = -\mathbf{1}$, and $\boldsymbol{\Sigma}_1$ and $\boldsymbol{\Sigma}_2$ are randomly generated positive definite matrices.

We aim to approximate the frequency by a 2nd-order expansion of our multivariate basis functions. Our optimization-based quadrature rule generates 33 pairs of quadrature samples and weights, then a deterministic periodic steady-state simulator is called repeatedly to simulate the oscillator at all parameter samples. Fig. 6 shows the obtained weights of all basis functions and the probability density function.

We compare the computed mean value from our methods with that from Monte Carlo in Table III. Monte Carlo method converges very slowly, and requires 3030× more simulation samples to achieve the similar level of accuracy (with 2 accurate fractional digits).

C. A Parallel Coupled Ring Resonator Optical Filter

In this subsection, we consider the 3-stage parallel-coupled ring resonator optical filter[1] in Fig. 5 (a). This optical filter is a versatile component for wavelength filtering, multiplexing, switching, and modulation in photonic integrated circuits. This circuit has a nominal 3-dB bandwidth of 12 GHz, and the coupling coefficients for the three rings are $K_1 = K_3 = 0.198836$ and $K_2 = 0.356423$. In the nominal design, the waveguide lengths $L_{12}, L_{21}, L_{23}, L_{32}$ are all 30.6624 μm, and the circumference of all ring are $R_1 = R_2 = R_3 = 2997.92$ μm. In

[1] The details of this benchmark can be found at https://kb.lumerical.com/en/pic_circuits_coupled_ring_resonator_filters.html

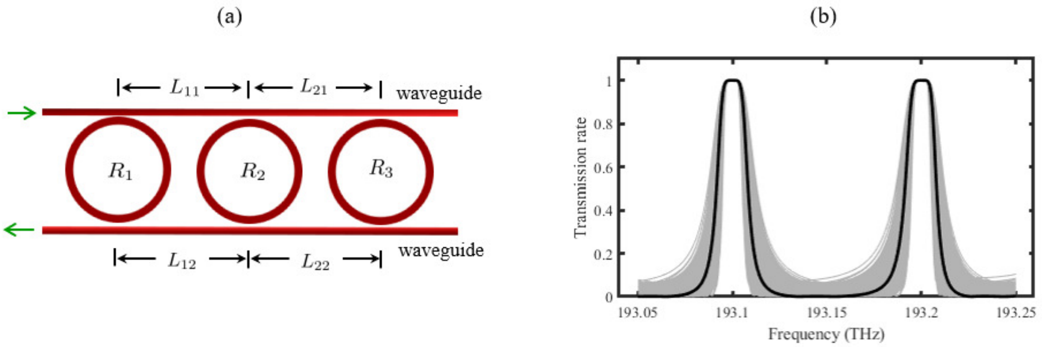


Fig. 5. (a) Schematic of a 3-stage parallel-coupled ring resonator optical filter. L_{12}, L_{21}, L_{23} and L_{32} are the connecting waveguides, and R_1, R_2 and R_3 denote the rings. (b) The black line shows the nominal transmission function, and the thin grey lines show the effect of fabrication uncertainties on the waveguide lengths of L_{12}, L_{21}, L_{23}, L_{32}.

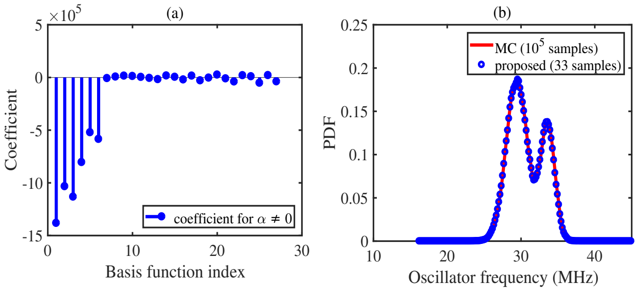


Fig. 6. Numerical results of the CMOS ring oscillator. (a) obtained coefficients/weights of our basis functions; (b) probability density functions of the oscillator frequency obtained by our proposed method and Monte Carlo (MC).

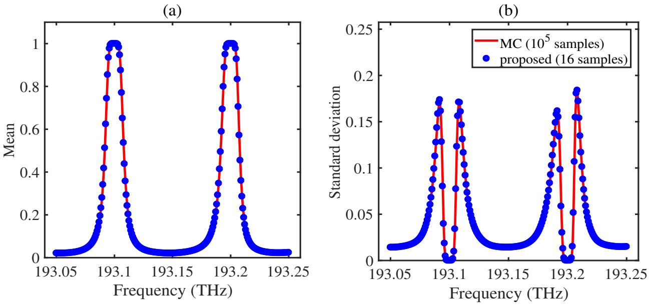


Fig. 7. Simulation results with respect to the geometric uncertainties in the waveguide length of L_{12}, L_{21}, L_{23} and L_{32}. (a) obtained mean value of the power transmission rate; (b) standard deviation of the transmission rate.

practice, there exist non-Gaussian correlated uncertainties in the waveguide geometric parameters. The effect of fabrication uncertainties are shown in Fig. 5 (b).

Our goal is to build a 2nd-order stochastic model to approximate the power transmission curve at different frequency points $y(f,\boldsymbol{\xi}) = \sum_{|\boldsymbol{\alpha}|=0}^{p} c_{\boldsymbol{\alpha}}(f)\Psi_{\boldsymbol{\alpha}}(\boldsymbol{\xi})$. We use a Gaussian mixture model to describe the uncertainties,

$$\boldsymbol{\xi} = \boldsymbol{\xi}_0 + \Delta\boldsymbol{\xi}, \text{ where } \Delta\boldsymbol{\xi} \sim \frac{1}{2}\mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) + \frac{1}{2}\mathcal{N}(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2).$$

For the waveguide length parameters, we use

$$\boldsymbol{\mu}_1 = -\boldsymbol{\mu}_2 = 25 \times \mathbf{1}\text{nm}, \; \boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2 = 6.25(\mathbf{I} + 0.5\mathbf{E}).$$

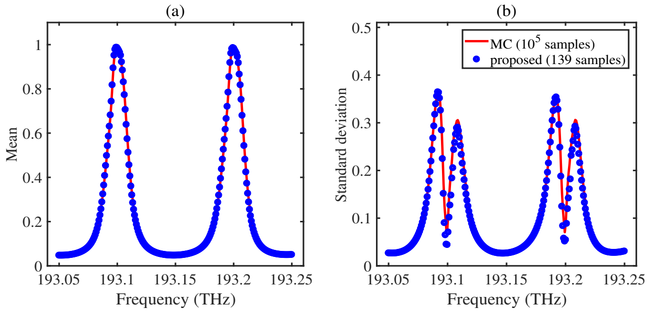


Fig. 8. Simulation results with respect to the geometric uncertainties in the waveguide length of L_{12}, L_{21}, L_{23}, L_{32}, R_1, R_2, R_3, and the uncertainties in effective index for L_{12}, L_{21}, L_{23} and L_{32}. (a) obtained mean value of the power transmission rate; (b) standard deviation of the transmission rate.

The uncertainties of the effective index follows a Gaussian mixture distribution with

$$\boldsymbol{\mu}_1 = -\boldsymbol{\mu}_2 = 10^{-3} \times \mathbf{1}, \; \boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2 = 10^{-6}(\mathbf{I} + 0.5\mathbf{E}).$$

We perform two experiments for the optical filter. The first experiment only considers the uncertainties of the waveguide lengths L_{12}, L_{21}, L_{23} and L_{32}. The second experiments considers uncertainties in the waveguide lengths L_{12}, L_{21}, L_{23}, ring geometry L_{32}, R_1, R_2 and R_3, as well as the effective index in L_{12}, L_{21}, L_{23} and L_{32}. The mean value and standard derivation of the output response are shown in Fig. 7 and Fig. 8, respectively. Although our method only uses 16 or 139 samples, it is able to achieve the similar accuracy with Monte Carlo that consumes 10^5 simulation samples.

D. An Arrayed Waveguide Grating (AWG)

Finally, we consider an arrayed waveguide grating (AWG) [64]. The AWG is essential for wavelength division and multiplexing in photonic systems. In our experiment, we use an AWGR with 9 waveguide arrays and two star couplers, as shown in Fig. 9 (a). In the nominal design, the radius of each star coupler is $R_1 = R_2 = 2.985$ mm, and the waveguide lengths $L_1, \ldots, L_9$ range from 46 μm to 420 μm. In practice, there exist non-Gaussian correlated uncertainties in the device geometric parameters, and the resulting performance uncertainties are shown in Fig. 9 (b).

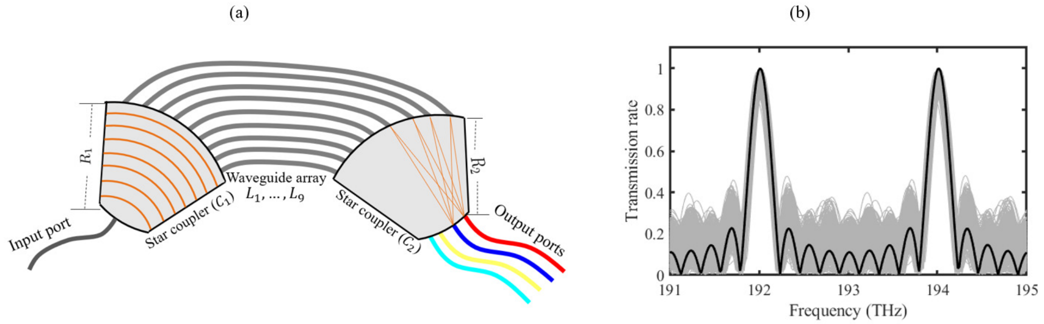


Fig. 9. (a) Schematic of an AWG with 9 waveguide arrays; (b) The nominal transmission rate from the input to output Port 1. The black curve shows the result without any uncertainties, and the grey lines show the effects caused by the fabrication uncertainties of radius R_1, R_2 and waveguide lengths $L_1, \ldots, L_9$.

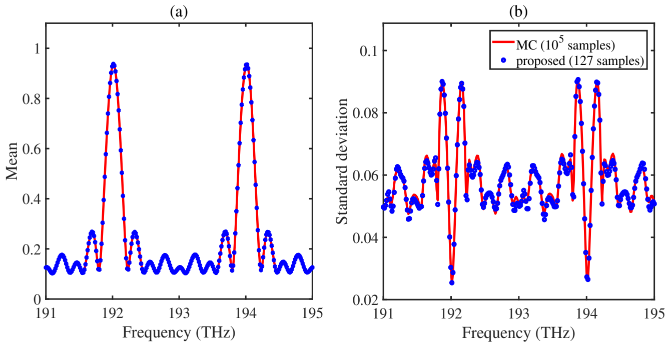


Fig. 10. Numerical results of the AWG with non-Gaussian correlated uncertainties in radius R_1, R_2 and the waveguide array lengths of $L_1, \ldots, L_9$. (a) mean value of the transmission rate; (b) standard deviation of the transmission rate obtained by our proposed method and Monte Carlo (MC).

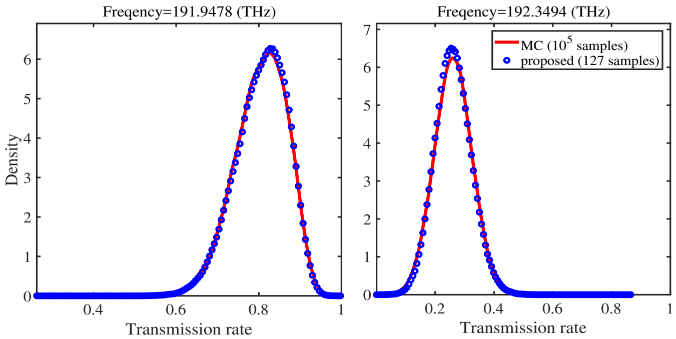


Fig. 11. Probability density functions of the transmission rates at two frequency points $f = 191.9478$ THz and $f = 192.3494$ THz obtained by our proposed method and Monte Carlo (MC)

We aim to build a 2nd-order stochastic model to approximate the transmission rates. A Gaussian-mixture model is used to describe the geometric uncertainties:

$$\boldsymbol{\xi} = \boldsymbol{\xi}_0 + \Delta\boldsymbol{\xi}, \text{ where } \Delta\boldsymbol{\xi} \sim \frac{1}{2}\mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) + \frac{1}{2}\mathcal{N}(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2).$$

For the radius of the star couplers, we set the mean values as $\boldsymbol{\mu}_1 = -\boldsymbol{\mu}_2 = 29.8 \times \mathbf{1}$ μm. For the waveguide array lengths, we set $\boldsymbol{\mu}_1 = -\boldsymbol{\mu}_2 = 0.05 \times \mathbf{1}$ μm. The covariance matrices are block diagonal positive definite.

We compare the computed mean value and standard deviation of our method with that from Monte Carlo in Fig. 10. Using only 127 simulation samples, our method is able to achieve the similar accuracy with 10^5 Monte Carlo samples. Fig. 11 further shows the probability density functions of the transmission rates at two frequency points $f = 191.9478$ THz and $f = 192.3494$ THz.

E. Practical Number of Quadrature Samples

Finally, Table IV shows the number of quadrature samples used by our approach in all numerical experiments. The lower and upper bounds of the number of samples from Theorem 1 are listed in the last two columns. Clearly, in most cases, the practical number of samples is very close to the lower bound. When the order of basis function is very high, the obtained number of quadrature samples may occasionally becomes close to the upper bound. This is because the following reason: when p is very large, the objective function in (11) is a polynomial function of extremely high order (i.e., $4p$), and the coordinate descent solver becomes hard to converge. We expect that the number of quadrature samples will also be close to the theoretical lower bound even for very large p, if a better nonlinear optimization solver is developed in the future.

TABLE IV

THE NUMBER OF QUADRATURE SAMPLES USED IN ALL EXPERIMENTS. HERE, p DENOTES THE MAXIMAL ORDER OF BASIS FUNCTIONS, d IS THE NUMBER OF RANDOM PARAMETERS.

			Proposed		
Benchmarks	p	d	# samples	lower bound	upper bound
Synthetic	1	2	3	3	6
	2	2	6	6	15
	3	2	10	10	28
	4	2	17	15	45
	5	2	66	21	66
CMOS ring	2	6	33	28	210
Optical filter	2	4	16	15	70
	2	11	139	78	1365
AWG	2	11	127	78	1365

VII. Conclusion and Remarks

This paper has investigated a long-standing research challenge: how can we handle non-Gaussian correlated uncertainties by stochastic spectral methods? We have proposed several theories and algorithms to overcome this challenge and have tested them by various benchmarks. Specifically, we have proposed a set of orthonormal basis functions that work extremely well for non-Gaussian correlated process variations which are beyond the capability of the existing well-known generalized polynomial-chaos theory. We have presented an optimization approach to calculate the quadrature nodes and weights required in the projection step. We have also provided some rigorous theoretical results regarding the required number of quadrature samples and the error bound of our framework. Our method has demonstrated a nearly exponential convergence rate on a smooth synthetic example. It has also achieved 700× to 6000× speedup than Monte Carlo on several practical design benchmarks, including a CMOS electronic ring oscillator, an optical filter built with 3-stage photonic ring resonators and an arrayed waiveguide grating.

We have two final remarks:

- Based on our theoretical analysis, we conclude that as long as the stochastic unknown output is smooth enough, and if the the optimization solver in our quadrature rule has a small error, both the numerical integration and approximation error will be very small, leading to highly accurate results in our stochastic collocation framework.
- It remains an open problem to determinate the required minimum number of quadrature nodes. Our numerical experiments show an excellent heuristic result: the practical number of quadrature nodes used in our framework is almost always close to the theoretical lower bound.

Acknowledgment

The authors would like to thank the anonymous reviewers for their detailed comments. We also appreciate Allen Sadun, Kaiqi Zhag and Kaikai Liu for their helpful discussions on the benchmarks and on Lumerical interconnect, and thank Max Gershman for his help on some of the code implementation.

Appendix A
Proof of Theorem 1

We show the lower bound and upper bound of the number of quadrature points required to achieve $2p$-th-order accuracy are N_p and N_{2p}, respectively.

Firstly, according to Appendix B, (29) holds if the quadrature points and weights satisfy (10). As a result, we have

$$\mathbf{Q}\text{diag}(\mathbf{w})\mathbf{Q}^T = \mathbf{I}_{N_p}, \tag{26}$$

where $\mathbf{Q} \in \mathbb{R}^{N_p \times M}$ with each element $\mathbf{Q}_{ij} = \Psi_i(\boldsymbol{\xi}_j)$, and $\mathbf{I}_{N_p}$ is an N_p-by-N_p identity matrix. Because the right-hand side is full rank, $\mathbf{Q}$ has a full row rank and thus $M \geq N_p$.

We further notice that the first row of (10) is $\sum_{k=1}^{M} w_k = 1$, therefore (10) can be rewritten as

$$\mathbf{Q}_1\mathbf{w} = \mathbf{0}_{N_{2p}-1},\ \sum_{k=1}^{M} w_k = 1,\ \mathbf{w} \geq 0, \tag{27}$$

where $\mathbf{Q}_1 \in \mathbb{R}^{(N_{2p}-1)\times M}$ consists of the last $N_{2p}-1$ rows of $\mathbf{Q}$, and $\mathbf{0}_{N_{2p}-1} \in \mathbb{R}^{N_{2p}-1}$ is a zero vector. According to the Carathéodory's Theorem [65], because $\mathbf{0}_{N_{2p}}$ lies in the convex hull formed by the column vectors of $\mathbf{Q}_1$, it can be written as the convex combination of not more than N_{2p} column vectors. In other words, there exists a matrix $\hat{\mathbf{Q}}_1$ formed by only N_{2p} columns of $\mathbf{Q}_1$ such that (27) still holds if we replace $\mathbf{Q}_1$ with $\hat{\mathbf{Q}}_1$ and change the length of $\mathbf{w}$ accordingly. Vector $\mathbf{0}_{N_{2p}-1}$ being in the convex hull of $\mathbf{Q}_1$ is a natural result of our numerical quadrature rule defined on the selected basis functions, therefore there exists $M \leq N_{2p}$.

Remark In the above proof, we show that by Carathéodory's Theorem, there exists N_{2p} quadrature nodes and weights such that (27) is true. In general, we do not know how to choose the N_{2p} sample nodes and weights *a priori*. However, our optimization solver can automatically calculate these quadrature nodes and weights. On the contrary, the linear programming approach in [58] needs to prescribe the sampling nodes and only calculate the weights, and it can not guarantee the conditions in (27).

Appendix B
Proof of Theorem 2

In order to show the exact recovery of $y(\boldsymbol{\xi}) \in \boldsymbol{\mathcal{S}}_p$, we need to prove that

$$c_{\boldsymbol{\alpha}} = \tilde{c}_{\boldsymbol{\alpha}},\ \forall\ |\boldsymbol{\alpha}| \leq p. \tag{28}$$

Here $\tilde{c}_{\boldsymbol{\alpha}}$ is obtained by the following numerical scheme:

$$\begin{aligned}\tilde{c}_{\boldsymbol{\alpha}} &= \sum_{k=1}^{M} y(\boldsymbol{\xi}_k)\Psi_{\boldsymbol{\alpha}}(\boldsymbol{\xi}_k)w_k = \sum_{k=1}^{M}\sum_{|\boldsymbol{\beta}|=0}^{p} c_{\boldsymbol{\beta}}\Psi_{\boldsymbol{\beta}}(\boldsymbol{\xi}_k)\Psi_{\boldsymbol{\alpha}}(\boldsymbol{\xi}_k)w_k \\ &= \sum_{|\boldsymbol{\beta}|=0}^{p} c_{\boldsymbol{\beta}}\left(\sum_{k=1}^{M}\Psi_{\boldsymbol{\beta}}(\boldsymbol{\xi}_k)\Psi_{\boldsymbol{\alpha}}(\boldsymbol{\xi}_k)w_k\right).\end{aligned}$$

A sufficient condition of (28) is

$$\sum_{k=1}^{M}\Psi_{\boldsymbol{\beta}}(\boldsymbol{\xi}_k)\Psi_{\boldsymbol{\alpha}}(\boldsymbol{\xi}_k)w_k = \delta_{\boldsymbol{\alpha},\boldsymbol{\beta}}. \tag{29}$$

In fact, the left-hand side of (29) is the numerical approximation for the integral $\mathbb{E}[\Psi_{\boldsymbol{\beta}}(\boldsymbol{\xi})\Psi_{\boldsymbol{\alpha}}(\boldsymbol{\xi})]$, which is guaranteed to be exact if we have a quadrature rule that can exactly evaluate the integration of every basis function bounded by order $2p$. In other words, (10) is a sufficient condition for (29).

Appendix C
Proof of Theorem 3

Before the detailed proof, we first introduce the Hölder's inequality [66] that will be used in our theoretical analysis.

- Hölder's inequality for the Euclidean vector space: for all vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ and $q_1, q_2 \in [1, +\infty]$ with $\frac{1}{q_1} + \frac{1}{q_2} = 1$,

$$\left|\sum_{i=1}^{n} x_i y_i\right| \leq \sum_{i=1}^{n} |x_i y_i| \leq \|\mathbf{x}\|_{q_1}\|\mathbf{y}\|_{q_2}. \tag{30}$$

For the special case $q_1 = 1$ and $q_2 = +\infty$, there is

$$\left|\sum_{i=1}^{n} x_i y_i\right| \leq \sum_{i=1}^{n} |x_i y_i| \leq \|\mathbf{x}\|_1\|\mathbf{y}\|_\infty. \tag{31}$$

- Hölder's inequality in the probability space: for all measurable functions $f(\boldsymbol{\xi})$ and $g(\boldsymbol{\xi})$ and $q_1, q_2 \in [1, +\infty]$ with $\frac{1}{q_1} + \frac{1}{q_2} = 1$:
$$\mathbb{E}[|f(\boldsymbol{\xi})g(\boldsymbol{\xi})|] \leq \|f(\boldsymbol{\xi})\|_{q_1} \|g(\boldsymbol{\xi})\|_{q_2}. \tag{32}$$
For the special case $g(\boldsymbol{\xi}) \equiv 1$ and $q_1 = q_2 = 2$, there is
$$\|f(\boldsymbol{\xi})\|_1 = \mathbb{E}[|f(\boldsymbol{\xi})|] \leq (\mathbb{E}[|f(\boldsymbol{\xi})|^2])^{\frac{1}{2}} = \|f(\boldsymbol{\xi})\|_2. \tag{33}$$

Now we start to prove Theorem 3. According to the definition $y_p(\boldsymbol{\xi}) = \sum_{|\boldsymbol{\alpha}|=0}^{p} c_{\boldsymbol{\alpha}} \Psi_{\boldsymbol{\alpha}}(\boldsymbol{\xi})$ and $c_{\boldsymbol{\alpha}} = \mathbb{E}[y(\boldsymbol{\xi})\Psi_{\boldsymbol{\alpha}}(\boldsymbol{\xi})]$, we have
$$\mathbb{E}[y(\boldsymbol{\xi})\Psi_j(\boldsymbol{\xi})] = \mathbb{E}[y_p(\boldsymbol{\xi})\Psi_j(\boldsymbol{\xi})] = c_j, \forall j = 1, \ldots, N_p. \tag{34}$$
We consider $j = 1$ and $\Psi_1(\boldsymbol{\xi}) = 1$, then the above equation indicates $\mathbb{E}[y(\boldsymbol{\xi})] = \mathbb{E}[y_p(\boldsymbol{\xi})] = c_0$. Based on this observation, we can estimate the difference between $\mathbb{E}[y(\boldsymbol{\xi})]$ and $\mathbb{I}[y(\boldsymbol{\xi})]$:
$$\begin{aligned} &|\mathbb{E}[y(\boldsymbol{\xi})] - \mathbb{I}[y(\boldsymbol{\xi})]| = |\mathbb{E}[y_p(\boldsymbol{\xi})] - \mathbb{I}[y(\boldsymbol{\xi})]| \\ \leq &\underbrace{|\mathbb{E}[y_p(\boldsymbol{\xi})] - \mathbb{I}[y_p(\boldsymbol{\xi})]|}_{\text{(a)}} + \underbrace{|\mathbb{I}[y_p(\boldsymbol{\xi})] - \mathbb{I}[y(\boldsymbol{\xi})]|}_{\text{(b)}}. \end{aligned} \tag{35}$$

Item (a) arises from the error of our numerical quadrature:
$$\begin{aligned} (a) = |\mathbb{E}[y_p(\boldsymbol{\xi})] - \mathbb{I}[y_p(\boldsymbol{\xi})]| = &\left| \sum_{j=1}^{N_p} c_j \left(\mathbb{E}[\Psi_j(\boldsymbol{\xi})] - \mathbb{I}[\Psi_j(\boldsymbol{\xi})] \right) \right| \\ \leq &\|\mathbf{c}\|_\infty \|\Phi(\bar{\boldsymbol{\xi}})\mathbf{w} - \mathbf{e}_1\|_1 \leq L\epsilon. \end{aligned} \tag{36}$$
The first inequality results from the Hölder's inequality (31). The second inequality follows from $\|\Phi(\bar{\boldsymbol{\xi}})\mathbf{w} - \mathbf{e}_1\|_1 \leq \epsilon$ in (21), and we have $\|\mathbf{c}\|_\infty \leq L$ because
$$|c_j| = |\mathbb{E}[y(\boldsymbol{\xi})\Psi_j(\boldsymbol{\xi})]| \leq \|y(\boldsymbol{\xi})\|_2 \|\Psi_j(\boldsymbol{\xi})\|_2 \leq L\|\Psi_j(\boldsymbol{\xi})\|_2 = L.$$

Item (b) is due to the projection error
$$\begin{aligned} (b) = &|\mathbb{I}[y_p(\boldsymbol{\xi}) - \mathbb{I}[y(\boldsymbol{\xi})]| \\ \leq &W\|y(\boldsymbol{\xi}) - y_p(\boldsymbol{\xi})\|_1 \leq W\|y(\boldsymbol{\xi}) - y_p(\boldsymbol{\xi})\|_2 \leq W\delta. \end{aligned} \tag{37}$$
The first inequality follows from that the operator $\mathbb{I}$ is bounded by W in (20). The second inequality results from the Hölder's inequality (33). The last inequality follows from our assumption $\|y(\boldsymbol{\xi}) - y_p(\boldsymbol{\xi})\|_2 \leq \delta$ in (18).

Combing (35), (36) and (37), we have
$$|\mathbb{E}[y(\boldsymbol{\xi})] - \mathbb{I}[y(\boldsymbol{\xi})]| \leq L\epsilon + W\delta. \tag{38}$$
The proof of Theorem 3 is complete.

Appendix D
Proof of Theorem 4

The total error of our stochastic collocation algorithm can be bounded by two terms:
$$\|y(\boldsymbol{\xi}) - \tilde{y}(\boldsymbol{\xi})\|_2 \leq \|y(\boldsymbol{\xi}) - y_p(\boldsymbol{\xi})\|_2 + \|y_p(\boldsymbol{\xi}) - \tilde{y}(\boldsymbol{\xi})\|_2.$$
Based on Assumption 2, the first item is upper bounded by δ. We only need to estimate the second term. In fact,
$$\|y_p(\boldsymbol{\xi}) - \tilde{y}(\boldsymbol{\xi})\|_2 = \|\sum_{j=1}^{N_p} (c_j - \tilde{c}_j)\Psi_j(\boldsymbol{\xi})\|_2 = \sqrt{\sum_{j=1}^{N_p} (c_j - \tilde{c}_j)^2},$$
where the last equality follows the fact that the chosen basis functions are orthogonal and normalized. Furthermore,
$$\begin{aligned} |c_j - \tilde{c}_j| = &|\mathbb{E}[y_p(\boldsymbol{\xi})\Psi_j(\boldsymbol{\xi})] - \mathbb{I}[y(\boldsymbol{\xi})\Psi_j(\boldsymbol{\xi})]| \\ \leq &\underbrace{|\mathbb{E}[y_p(\boldsymbol{\xi})\Psi_j(\boldsymbol{\xi})] - \mathbb{I}[y_p(\boldsymbol{\xi})\Psi_j(\boldsymbol{\xi})]|}_{(a)} \qquad (39) \\ &+ \underbrace{|\mathbb{I}[(y_p(\boldsymbol{\xi}) - y(\boldsymbol{\xi}))\Psi_j(\boldsymbol{\xi})]|}_{(b)}. \qquad (40) \end{aligned}$$

Both $y_p(\boldsymbol{\xi})$ and $\Psi_j(\boldsymbol{\xi})$ are polynomials bounded by order p, so their product is a polynomial bounded by order $2p$, i.e., $y_p(\boldsymbol{\xi})\Psi_j(\boldsymbol{\xi}) \in \mathcal{S}_{2p}$. There exists an expansion $y_p(\boldsymbol{\xi})\Psi_j(\boldsymbol{\xi}) = \sum_{l=1}^{N_{2p}} a_l \Psi_l(\boldsymbol{\xi})$ and an upper bound for term (a):
$$\begin{aligned} (a) = &|\mathbb{E}[y_p(\boldsymbol{\xi})\Psi_j(\boldsymbol{\xi})] - \mathbb{I}[y_p(\boldsymbol{\xi})\Psi_j(\boldsymbol{\xi})]| \\ = &\left| \sum_{l=1}^{N_{2p}} a_l \left(\mathbb{E}[\Psi_l(\boldsymbol{\xi})] - \mathbb{I}[\Psi_l(\boldsymbol{\xi})] \right) \right| \\ \leq &\|\mathbf{a}\|_\infty \|\Phi(\bar{\boldsymbol{\xi}})\mathbf{w} - \mathbf{e}_1\|_1 \leq LT\epsilon. \end{aligned} \tag{41}$$
The first inequality is due to (31), and the last inequality follows from
$$a_l = \mathbb{E}[y_p(\boldsymbol{\xi})\Psi_j(\boldsymbol{\xi})\Psi_l(\boldsymbol{\xi})] \leq \|y_p(\boldsymbol{\xi})\|_2 \|\Psi_j(\boldsymbol{\xi})\Psi_l(\boldsymbol{\xi})\|_2 \leq LT \tag{42}$$
where $T = \max_{j,l=1,\ldots,N_{2p}} \|\Psi_j(\boldsymbol{\xi})\Psi_l(\boldsymbol{\xi})\|_2$.

We can also find an upper bound for term (b) in (40):
$$\begin{aligned} (b) = &|\mathbb{I}[(y_p(\boldsymbol{\xi}) - y(\boldsymbol{\xi}))\Psi_j(\boldsymbol{\xi})]| \\ \leq &W\|(y_p(\boldsymbol{\xi}) - y(\boldsymbol{\xi}))\Psi_j(\boldsymbol{\xi})\|_1 \\ \leq &W\|(y_p(\boldsymbol{\xi}) - y(\boldsymbol{\xi}))\|_2 \|\Psi_j(\boldsymbol{\xi})\|_2 \\ = &W\|(y_p(\boldsymbol{\xi}) - y(\boldsymbol{\xi}))\|_2 \leq W\delta. \end{aligned} \tag{43}$$
Combing (39), (40), (41) and (43), we have $|c_j - \tilde{c}_j| \leq LT\epsilon + W\delta$, and thus $\|y_p(\boldsymbol{\xi}) - \tilde{y}(\boldsymbol{\xi})\|_2 \leq N_p(LT\epsilon + W\delta)$. Noting that $\|y(\boldsymbol{\xi}) - y_p(\boldsymbol{\xi})\|_2 \leq \delta$, we finally have
$$\|y(\boldsymbol{\xi}) - \tilde{y}(\boldsymbol{\xi})\|_2 \leq \delta + N_p(LT\epsilon + W\delta). \tag{44}$$
This completes the proof of Theorem 4.

Remark To show a_l is bounded in (42) is equivalent to show $\boldsymbol{\mathcal{S}}_{2p}$ is complete under the Minkowski sum, i.e.,
$$\boldsymbol{\mathcal{S}}_p \oplus \boldsymbol{\mathcal{S}}_p \subset \boldsymbol{\mathcal{S}}_{2p}. \tag{45}$$
In other words, if $p_1(\boldsymbol{\xi}), p_2(\boldsymbol{\xi}) \in \boldsymbol{\mathcal{S}}_p$, then $p_1(\boldsymbol{\xi})p_2(\boldsymbol{\xi}) \in \boldsymbol{\mathcal{S}}_{2p}$. Intuitively, this is true because the product of two p-th order polynomial is a polynomial bounded by order $2p$. A sufficient condition for (45) is that $\|\Psi_j(\boldsymbol{\xi})\Psi_l(\boldsymbol{\xi})\|_2$ is bounded. In real applications, most widely used distributions include Gaussian, Gaussian mixture distribution, or a distribution on a bounded domain can guarantee that the high-order moments are bounded. As a result, (45) holds in most cases. But there exists some rare density functions whose high-order moments are not necessarily bounded, such as the log norm distribution. In this rare case, the error analysis in Theorem 4 may not hold.

Appendix E
Proof for Lemma 1

In order to upper bound $\|\mathbf{V}-\mathbf{I}_{N_p}\|_F$, we consider the error for each element $\mathbb{E}[\Psi_i(\boldsymbol{\xi})\Psi_j(\boldsymbol{\xi})]-\mathbb{I}[\Psi_i(\boldsymbol{\xi})\Psi_j(\boldsymbol{\xi})]$. We can have an expansion $\Psi_i(\boldsymbol{\xi})\Psi_j(\boldsymbol{\xi})=\sum_{l=1}^{N_{2p}} a_l\Psi_l(\boldsymbol{\xi})$, then

$$
\begin{aligned}
&|\mathbb{E}[\Psi_i(\boldsymbol{\xi})\Psi_j(\boldsymbol{\xi})]-\mathbb{I}[\Psi_i(\boldsymbol{\xi})\Psi_j(\boldsymbol{\xi})]| \\
=&\left|\sum_{l=1}^{N_{2p}} a_l\left(\mathbb{E}[\Psi_l(\boldsymbol{\xi})]-\mathbb{I}[\Psi_l(\boldsymbol{\xi})]\right)\right| \\
\leq&\|\mathbf{a}\|_2\|\Phi(\bar{\boldsymbol{\xi}})\mathbf{w}-\mathbf{e}_1\|_2.
\end{aligned}
$$

Because $\|\mathbf{a}\|_2^2=\|\Psi_i(\boldsymbol{\xi})\Psi_j(\boldsymbol{\xi})\|_2^2\leq T^2$ and

$$\|\Phi(\bar{\boldsymbol{\xi}})\mathbf{w}-\mathbf{e}_1\|_2\leq\|\Phi(\bar{\boldsymbol{\xi}})\mathbf{w}-\mathbf{e}_1\|_1\leq\epsilon,$$

we have

$$|\mathbb{E}[\Psi_i(\boldsymbol{\xi})\Psi_j(\boldsymbol{\xi})]-\mathbb{I}[\Psi_i(\boldsymbol{\xi})\Psi_j(\boldsymbol{\xi})]|\leq T\epsilon,$$

and further obtain $\|\mathbf{V}-\mathbf{I}_{N_p}\|_F\leq N_pT\epsilon$.

References

[1] C. Cui, M. Gershman, and Z. Zhang, "Stochastic collocation with non-Gaussian correlated parameters via a new quadrature rule," in *Proc. IEEE Conf. EPEPS*. San Jose, CA, Oct. 2018, pp. 57–59.

[2] D. S. Boning, "Variation," *IEEE Trans. Semiconductor Manufacturing*, vol. 21, no. 1, pp. 63–71, Feb 2008.

[3] M. Miranda, "The threat of semiconductor variability," in *IEEE Spectrum*, June 2012.

[4] S. K. Selvaraja, W. Bogaerts, P. Dumon, D. Van Thourhout, and R. Baets, "Subnanometer linewidth uniformity in silicon nanophotonic waveguide devices using CMOS fabrication technology," *IEEE Journal of Selected Topics in Quantum Electronics*, vol. 16, no. 1, pp. 316–324, 2010.

[5] W. Zortman, D. Trotter, and M. Watts, "Silicon photonics manufacturing," *Opt. Express*, vol. 18, no. 23, pp. 23 598–23 607, Nov 2010.

[6] L. Chrostowski, X. Wang, J. Flueckiger, Y. Wu, Y. Wang, and S. T. Fard, "Impact of fabrication non-uniformity on chip-scale silicon photonic integrated circuits," in *Optical Fiber Communication Conf.*, 2014, pp. Th2A–37.

[7] Z. Lu, J. Jhoja, J. Klein, X. Wang, A. Liu, J. Flueckiger, J. Pond, and L. Chrostowski, "Performance prediction for silicon photonics integrated circuits with layout-dependent correlated manufacturing variability," *Optics Express*, vol. 25, no. 9, pp. 9712–9733, 2017.

[8] S. Weinzierl, "Introduction to Monte Carlo methods," NIKHEF, Theory Group, The Netherlands, Tech. Rep. NIKHEF-00-012, 2000.

[9] D. Xiu, *Numerical Methods for Stochastic Computations: A Spectral Method Approach*. Princeton University Press, 2010.

[10] D. Xiu and G. E. Karniadakis, "The Wiener-Askey polynomial chaos for stochastic differential equations," *SIAM J. Sci. Comp.*, vol. 24, no. 2, pp. 619–644, Feb 2002.

[11] R. Ghanem and P. Spanos, *Stochastic finite elements: a spectral approach*. Springer-Verlag, 1991.

[12] Z. Zhang, T. A. El-Moselhy, I. A. M. Elfadel, and L. Daniel, "Stochastic testing method for transistor-level uncertainty quantification based on generalized polynomial chaos," *IEEE Trans. Computer-Aided Design Integr. Circuits Syst.*, vol. 32, no. 10, pp. 1533–1545, Oct. 2013.

[13] D. Xiu and J. S. Hesthaven, "High-order collocation methods for differential equations with random inputs," *SIAM J. Sci. Comp.*, vol. 27, no. 3, pp. 1118–1139, Mar 2005.

[14] S. Vrudhula, J. M. Wang, and P. Ghanta, "Hermite polynomial based interconnect analysis in the presence of process variations," *IEEE Trans. CAD of Integr. Circuits Syst.*, vol. 25, no. 10, pp. 2001–2011, 2006.

[15] T.-A. Pham, E. Gad, M. S. Nakhla, and R. Achar, "Decoupled polynomial chaos and its applications to statistical analysis of high-speed interconnects," *IEEE Transactions on Components, Packaging and Manufacturing Technology*, vol. 4, no. 10, pp. 1634–1647, 2014.

[16] I. S. Stievano, P. Manfredi, and F. G. Canavero, "Parameters variability effects on multiconductor interconnects via hermite polynomial chaos," *IEEE Trans. Compon., Packag., Manufacut. Tech.*, vol. 1, no. 8, pp. 1234–1239, Aug. 2011.

[17] K. Strunz and Q. Su, "Stochastic formulation of SPICE-type electronic circuit simulation with polynomial chaos," *ACM Trans. Modeling and Computer Simulation*, vol. 18, no. 4, pp. 15:1–15:23, Sep 2008.

[18] R. Pulch, "Modelling and simulation of autonomous oscillators with random parameters," *Mathematics and Computers in Simulation*, vol. 81, no. 6, pp. 1128–1143, Feb 2011.

[19] M. Rufuie, E. Gad, M. Nakhla, R. Achar, and M. Farhan, "Fast variability analysis of general nonlinear circuits using decoupled polynomial chaos," in *Workshop Signal and Power Integrity*, May 2014, pp. 1–4.

[20] P. Manfredi, D. V. Ginste, D. D. Zutter, and F. Canavero, "Stochastic modeling of nonlinear circuits via SPICE-compatible spectral equivalents," *IEEE Trans. Circuits Syst. I: Regular Papers*, vol. 61, no. 7, pp. 2057–2065, July 2014.

[21] Z. Zhang, T. A. El-Moselhy, P. Maffezzoni, I. A. M. Elfadel, and L. Daniel, "Efficient uncertainty quantification for the periodic steady state of forced and autonomous circuits," *IEEE Trans. Circuits Syst. II: Exp. Briefs*, vol. 60, no. 10, Oct. 2013.

[22] M. Ahadi and S. Roy, "Sparse linear regression (SPLINER) approach for efficient multidimensional uncertainty quantification of high-speed circuits." *IEEE Trans. on CAD of Integrated Circuits and Systems*, vol. 35, no. 10, pp. 1640–1652, 2016.

[23] A. C. Yucel, H. Bagci, and E. Michielssen, "An ME-PC enhanced HDMR method for efficient statistical analysis of multiconductor transmission line networks," *IEEE Trans. Components, Packaging and Manufacturing Technology*, vol. 5, no. 5, pp. 685–696, May 2015.

[24] Z. Zhang, X. Yang, G. Marucci, P. Maffezzoni, I. M. Elfadel, G. Karniadakis, and L. Daniel, "Stochastic testing simulator for integrated circuits and MEMS: Hierarchical and sparse techniques," in *Proc. IEEE Custom Integrated Circuits Conf.* San Jose, CA, Sept. 2014, pp. 1–8.

[25] Z. Zhang, I. Osledets, X. Yang, G. E. Karniadakis, and L. Daniel, "Enabling high-dimensional hierarchical uncertainty quantification by ANOVA and tensor-train decomposition," *IEEE Trans. CAD of Integrated Circuits and Systems*, vol. 34, no. 1, pp. 63 – 76, Jan 2015.

[26] T.-W. Weng, Z. Zhang, Z. Su, Y. Marzouk, A. Melloni, and L. Daniel, "Uncertainty quantification of silicon photonic devices with correlated and non-Gaussian random parameters," *Optics Express*, vol. 23, no. 4, pp. 4242 – 4254, Feb 2015.

[27] A. Waqas, D. Melati, P. Manfredi, and A. Melloni, "Stochastic process design kits for photonic circuits based on polynomial chaos augmented macro-modelling," *Optics Express*, vol. 26, no. 5, pp. 5894–5907, 2018.

[28] Z. Zhang, T.-W. Weng, and L. Daniel, "Big-data tensor recovery for high-dimensional uncertainty quantification of process variations," *IEEE Transactions on Components, Packaging and Manufacturing Technology*, vol. 7, no. 5, pp. 687–697, 2017.

[29] X. Li, "Finding deterministic solution from underdetermined equation: large-scale performance variability modeling of analog/rf circuits," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 29, no. 11, pp. 1661–1668, 2010.

[30] X. Yang, M. Choi, G. Lin, and G. E. Karniadakis, "Adaptive ANOVA decomposition of stochastic incompressible and compressible flows," *Journal of Computational Physics*, vol. 231, no. 4, pp. 1587–1614, 2012.

[31] X. Ma and N. Zabaras, "An adaptive high-dimensional stochastic model representation technique for the solution of stochastic partial differential equations," *Journal of Computational Physics*, vol. 229, no. 10, pp. 3884–3915, 2010.

[32] T. El-Moselhy and L. Daniel, "Stochastic dominant singular vectors method for variation-aware extraction," in *Proc. Design Automation Conference*, 2010, pp. 667–672.

[33] ——, "Variation-aware interconnect extraction using statistical moment preserving model order reduction," in *Proc. Design, Automation and Test in Europe*, 2010, pp. 453–458.

[34] J. Pond, J. Klein, J. Flückiger, X. Wang, Z. Lu, J. Jhoja, and L. Chrostowski, "Predicting the yield of photonic integrated circuits using statistical compact modeling," in *Integrated Optics: Physics and Simulations III*, vol. 10242, 2017, p. 102420S.

[35] H. Chang and S. S. Sapatnekar, "Statistical timing analysis considering spatial correlations using a single PERT-like traversal," in *Proc. Intl. Conference on Computer-Aided Design*, 2003, pp. 621–625.

[36] S. Bhardwaj, S. Vrudhula, P. Ghanta, and Y. Cao, "Modeling of intra-die process variations for accurate analysis and optimization of nano-scale circuits," in *Proc. Design Automation Conference*, 2006, pp. 791–796.

[37] S. S. Sapatnekar, "Overcoming variations in nanometer-scale technologies," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 1, no. 1, pp. 5–18, 2011.

[38] M. Rosenblatt, "Remarks on a multivariate transformation," *The annals of mathematical statistics*, vol. 23, no. 3, pp. 470–472, 1952.

[39] C. Soize and R. Ghanem, “Physical systems with random uncertainties: chaos representations with arbitrary probability measure,” *SIAM Journal on Scientific Computing*, vol. 26, no. 2, pp. 395–410, 2004.

[40] G. H. Golub and J. H. Welsch, “Calculation of Gauss quadrature rules,” *Math. Comp.*, vol. 23, pp. 221–230, 1969.

[41] Y. Xu, “Optimal points for cubature rules and polynomial interpolation on a square,” in *Contemporary Computational Mathematics-A Celebration of the 80th Birthday of Ian Sloan*. Springer, 2018, pp. 1287–1305.

[42] J. Paulson, E. Buehler, and A. Mesbah, “Arbitrary polynomial chaos for uncertainty propagation of correlated random variables in dynamic systems,” *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 3548–3553, 2017.

[43] M. Navarro, J. Witteveen, and J. Blom, “Polynomial chaos expansion for general multivariate distributions with correlated variables,” *arXiv preprint arXiv:1406.5483*, 2014.

[44] Z. Liu and Y. Choe, “Data-driven sensitivity indices for models with dependent inputs using the polynomial chaos expansion,” *arXiv preprint arXiv:1803.10978*, 2018.

[45] F. Nobile, R. Tempone, and C. G. Webster, “A sparse grid stochastic collocation method for partial differential equations with random input data,” *SIAM J. Numerical Analysis*, vol. 46, no. 5, pp. 2309–2345, 2008.

[46] D. Xiu and J. S. Hesthaven, “High-order collocation methods for differential equations with random inputs,” *SIAM Journal on Scientific Computing*, vol. 27, no. 3, pp. 1118–1139, 2005.

[47] I. Babuška, F. Nobile, and R. Tempone, “A stochastic collocation method for elliptic partial differential equations with random input data,” *SIAM J. Numer. Anal.*, vol. 45, no. 3, pp. 1005–1034, Mar 2007.

[48] F. Nobile, R. Tempone, and C. G. Webster, “A sparse grid stochastic collocation method for partial differential equations with random input data,” *SIAM J. Numer. Anal.*, vol. 46, no. 5, pp. 2309–2345, May 2008.

[49] ——, “An anisotropic sparse grid stochastic collocation method for partial differential equations with random input data,” *SIAM J. Numer. Anal.*, vol. 46, no. 5, pp. 2411–2442, May 2008.

[50] W. Gautschi, “On generating orthogonal polynomials,” *SIAM Journal on Scientific and Statistical Computing*, vol. 3, no. 3, pp. 289–317, 1982.

[51] T. Gerstner and M. Griebel, “Numerical integration using sparse grids,” *Numer. Algor.*, vol. 18, pp. 209–232, Mar. 1998.

[52] H. Zhu, X. Zeng, W. Cai, J. Xue, and D. Zhou, “A sparse grid based spectral stochastic collocation method for variations-aware capacitance extraction of interconnects under nanometer process technology,” in *Proc. Design Auto. Test in Europe*, Apr. 2007, pp. 1–6.

[53] V. Barthelmann, E. Novak, and K. Ritter, “High dimensional polynomial interpolation on sparse grids,” *Adv. Comput. Math.*, vol. 12, no. 4, pp. 273–288, Mar. 2000.

[54] Y. Xu, “On multivariate orthogonal polynomials,” *SIAM Journal Math. Analysis*, vol. 24, no. 3, pp. 783–794, 1993.

[55] R. Barrio, J. M. Pena, and T. Sauer, “Three term recurrence for the evaluation of multivariate orthogonal polynomials,” *Journal of Approximation Theory*, vol. 162, no. 2, pp. 407–420, 2010.

[56] G. H. Golub and C. F. Van Loan, *Matrix computations*. JHU Press, 2012, vol. 3.

[57] C. Cui and Z. Zhang, “Uncertainty quantification of electronic and photonic ICs with non-Gaussian correlated process variations,” in *Proc. Intl. Conf. Computer-Aided Design*. San Diego, CA, Nov. 2018, pp. 1–8.

[58] E. K. Ryu and S. P. Boyd, “Extensions of Gauss quadrature via linear programming,” *Foundations of Computational Mathematics*, vol. 15, no. 4, pp. 953–971, 2015.

[59] V. Keshavarzzadeh, R. M. Kirby, and A. Narayan, “Numerical integration in multiple dimensions with designed quadrature,” *SIAM Journal on Scientific Computing*, vol. 40, no. 4, pp. 2033–2061, 2018.

[60] R. Tappenden, P. Richtárik, and J. Gondzio, “Inexact coordinate descent: complexity and preconditioning,” *Journal of Optimization Theory and Applications*, vol. 170, no. 1, pp. 144–176, 2016.

[61] J. B. Lasserre, “Global optimization with polynomials and the problem of moments,” *SIAM J. Optimization*, vol. 11, no. 3, pp. 796–817, 2001.

[62] J. Nocedal and S. J. Wright, “Numerical optimization,” *Springer series in operations research and financial engineering*, 2006.

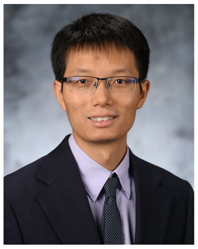
[63] A. K. Jain, M. N. Murty, and P. J. Flynn, “Data clustering: a review,” *ACM computing surveys (CSUR)*, vol. 31, no. 3, pp. 264–323, 1999.

[64] K. Zhang, X. Xiao, Y. Zhang, and S. B. Yoo, “Verilog-A compact modeling and simulation of AWGR based all-to-all optical interconnects,” in *CLEO: QELS_Fundamental Science*. OSA, 2018, pp. JW2A–49.

[65] I. Bárány and R. Karasev, “Notes about the carathéodory number,” *Discrete & Computational Geometry*, vol. 48, no. 3, pp. 783–792, 2012.

[66] K. Kuttler, “An introduction to linear algebra,” *Brigham Young University*, 2007.

Chunfeng Cui received the Ph.D. degree in computational mathematics from Chinese Academy of Sciences, Beijing, China, in 2016 with a specialization in numerical optimization. From 2016 to 2017, she was a Postdoctoral Fellow at City University of Hong Kong, Hong Kong. In 2017 She joined the Department of Electrical and Computer Engineering at University of California Santa Barbara as a Postdoctoral Scholar. From 2011 her research activity is mainly focused in the areas of tensor analysis and its applications. She has been working on numerical optimization algorithms for tensor problems, and its applications for machine learning and for uncertainty quantification of nano-scale chip design. She received the Best Paper Award of the IEEE EPEPS 2018.



Zheng Zhang (M'15) received his Ph.D degree in Electrical Engineering and Computer Science from the Massachusetts Institute of Technology (MIT), Cambridge, MA, in 2015. He is an Assistant Professor of Electrical and Computer Engineering with the University of California at Santa Barbara (UCSB), CA. His research interests include uncertainty quantification with applications to the design automation of multi-domain systems (e.g., nano-scale electronics, integrated photonics, and autonomous systems), and tensor computational methods for high-dimensional data analytics. His industrial experiences include Coventor Inc. and Maxim-IC; academic visiting experiences include UC San Diego, Brown University and Politechnico di Milano; government lab experiences include Argonne National Labs.

Dr. Zhang received the Best Paper Award of IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems in 2014, the Best Paper Award of IEEE Transactions on Components, Packaging and Manufacturing Technology in 2018, two Best Paper Awards (IEEE EPEPS 2018 and IEEE SPI 2016) and three additional Best Paper Nominations (CICC 2014, ICCAD 2011 and ASP-DAC 2011) at international conferences. His PhD dissertation was recognized by the ACM SIGDA Outstanding Ph.D Dissertation Award in Electronic Design Automation in 2016, and by the Doctoral Dissertation Seminar Award (i.e., Best Thesis Award) from the Microsystems Technology Laboratory of MIT in 2015. He was a recipient of the Li Ka-Shing Prize from the University of Hong Kong in 2011.