OPEN ACCESS
IOP Publishing | ISECAEP
Journal of Reliability Science and Engineering
J. Reliab. Sci. Eng. **2** (2026) 025002 (25pp)
https://doi.org/10.1088/3050-2454/ae5966

Automatic generation of high-dimensional surrogate models using moment quadrature design points

Xuefei Guan*

Graduate School of China Academy of Engineering Physics, Beijing 100193, People's Republic of China

E-mail: xfguan@gscaep.ac.cn

Received 20 November 2025, revised 26 February 2026
Accepted for publication 31 March 2026
Published 10 April 2026

Abstract

This study proposes a deterministic and definitive methodology for surrogate model construction based on moment quadrature design points. Different from adaptive or randomly sampling approaches, the proposed framework derives design points directly from the statistical moments of the input probability density functions, ensuring reproducibility, optimal polynomial exactness, and broad applicability to low- and high-dimensional problems involving arbitrary distributions, including nonuniform distributions and, in principle, empirical ones. It features an automatic pipeline for basis generation, coefficient regression, and optional term pruning, avoiding case-dependent tuning. Moreover, it allows direct evaluation of uncertainty metrics and global sensitivity indices to be obtained using the same set of model evaluations. Extensive numerical examples, including standard benchmark functions and engineering applications involving up to 100 random variables, demonstrate that the proposed method achieves high accuracy and computational efficiency compared with conventional surrogate modeling approaches.

Keywords: surrogate model, high-dimensional, moment quadrature method, design point

1. Introduction

Surrogate modeling plays a key role in the field of risk and reliability, and it has increasingly been emphasized with the ever-increasing need for complex system modeling and predictive maintenance. As engineering models become more complex, such as digital twins and large simulation models, the computational demand poses a great challenge to uncertainty quantification and analysis for those models spanning from classical to quantum regimes [1]. Therefore, constructing optimal surrogate models with minimal computational overhead is of paramount importance.

The surrogate modeling methodologies can loosely be classified into several categories. Polynomial-based surrogate modeling methods are widely used, such as classical polynomial response surface models and orthogonal polynomials under prescribed probability density functions (PDFs). In particular, the polynomial chaos expansion (PCE) has gained significant attention due to the advantage of extracting the mean and variance of the response, as well as Sobol' sensitivity indices from the resulting coefficients of the orthogonal polynomials [2–4]. Recent work on PCE surrogate models is largely centered on using learning-based techniques to extend the scope and applicability of the classical PCE, e.g. manifold learning-based PCE for high-dimensional surrogate models [5], deep kernel-PCE for high dimensional uncertainty analysis [6], physics-constrained machine learning PCE for uncertainty quantification [7], and various applications in different fields [8–11], to name a few.

Another surrogate family is the Gaussian process (GP) models in which the response of a physical process is modeled by a multivariate normal distribution. GP modeling is also

* Author to whom any correspondence should be addressed.

known as Kriging in some fields. When both the mean vector of the GP model and the auto-covariance function are known, the model is also referred to as simple Kriging. When the mean vector is unknown but constant and the auto-covariance function is known, the model is referred to as ordinary Kriging; while for a GP model with unknown mean vector in the form of a linear interpolator and a known auto-covariance function, the model is the so-called universal Kriging [12]. Evidently, the linear interpolator in universal Kriging can be a regular polynomial representation or a PCE. Recent developments in GP surrogate modeling includes the adaptive approach for incorporating partial information through Bayesian updating [13], new parameter estimation algorithms [14], the combination of a physics-based simulator with GP models [15], and new applications involving GP models [16, 17].

The other type of surrogate models for simulating stochastic processes is spectral representation-based methods, such as Karhune-Loève expansion, principal component analysis, and other orthogonal expansions aiming to minimize the mean squared errors between the resulting surrogate and the actual models [18]. Machine learning-based surrogate models have also gained increasing popularity due to the advancement of deep learning and large data [19–21]; nevertheless, those models normally require a sufficiently large amount of training data and the extrapolation of the built model to an unknown data space can be risky in realistic engineering problems due to the limited use of relevant physical laws [22].

While these different versatile approaches are valuable in terms of emulating the actual physical process using simple terms, from the mathematical perspective they are mainly different in the basis functions used for the reconstruction of the model responses at the design points, as well as the nature of the fitting coefficients, e.g. either deterministic or probabilistic, for different engineering purposes. Hence, the strategy of generating these design points and the results can drastically affect the effectiveness of the final model and the generality of the model prediction. Random generation of design points has been widely accepted as a practice to explore the space of input variables, and such a treatment can encode certain randomness. Latin hypercube sampling (LHS) is an efficient one-shot space-filling technique to generate a set of design points. Certain criteria can be adopted to enhance the space-filling property of the resulting samples, e.g. using a minimal distance constraint on any two samples. A recent comparative review on LHS with an adaptive sampling strategy particularly designed for Kriging can be found in [23]. However, as noted in [23], for a high-dimensional input space, such a treatment may not be very useful due to the curse of dimensionality. In addition, training a Kriging surrogate model directly in high dimensions can be more complex or even intractable, and dimension reduction techniques must be applied to make the training process manageable [24, 25]. The adaptive sampling methods reviewed above and other adaptive sampling methods bearing a similar idea, such as mode pursuing sampling and its variants [26–28], are not restricted to Kriging and radial basis function interpolators. The basic mechanism consists of an initial small random set of points, a score function to test the performance of a sample which is normally based on an intermediate surrogate constructed from the initial set of points, and an enrichment scheme based on the score function; hence, it is clear that the initial random set of points is of vital importance as it inherently determines the overall performance, i.e. the skeleton of the shape of the manifold, and the remaining part of those algorithms is essentially enriching local features of the skeleton. In fact, to minimize the influence of the initial small random set of points, multiple parallel runs are often adopted in those methods to enhance robustness. From the statistical point of view, such a treatment can loosely be equivalent to using a larger set of samples as an aggregate of multiple small sets for surrogate model building, which still poses a great challenge in high computational demanding cases. Another potential limitation of the adaptive methods is that there is no universal rule for selecting the score functions and criteria for generating new samples, introducing unavoidable arbitrariness and uncertainties into the final models that can sometimes be intractable or highly nontrivial to quantify.

In summary, current surrogate modeling methods still face the following challenges in resolving realistic complex black-box computational models. For one thing, a universal rule in generating the design points stemming from a rational measure to assess whether a set of design points is preferable or not appears to be absent, albeit such a rule is of paramount importance in all the subsequent surrogate modeling regardless of the basis functions chosen. For another, the construction and reduction of the surrogate models based on the generated design points for high-dimensional problems can be highly nontrivial for certain basis functions such as GP/Kriging models as emphasized in [23] and several others [29–33] as a result of the sophisticated and case-dependent tuning. The moment-based uncertainty and reliability analysis methods have also been discussed in [34] where a point-mapping-based sparse grid integration method to improve moment estimation efficiency in structural reliability analysis is proposed, in [35] where a Bayesian inference-assisted moment-based framework for reliability assessment of robotic systems under uncertainty is developed, and in [36] where a Bayesian updating method combined with dimension-reduction integration to efficiently evaluate multi-mode system failure probabilities is introduced. These studies primarily focus on moment estimation and reliability inference, while the present work emphasizes deterministic surrogate construction, where surrogate modeling, uncertainty propagation, and sensitivity analysis are integrated into a unified moment quadrature design point (MQDP)-based framework.

Motivated by the need for technical improvement towards resolving the aforementioned challenges, the goal of this study is to develop a deterministic and definitive methodology for surrogate model construction. By deterministic it is implied that there is no case-dependent tuning process as required by

many adaptive methods, and by definitive, it means that the overall procedure is mechanistic, and a streamlined procedure can be formulated for arbitrary black-box computational models. To achieve the first desired feature, a design point generation scheme is proposed based on the moment quadrature method, and it can be mathematically shown that when the polynomial basis functions, whether orthogonal or not, are used, the design points are optimal in terms of approximation exactness. For the second desired feature, ensuring that the surrogate model can always be established, regular polynomial terms are automatically enumerated according to the tensor product of the generated design points, and a linear superposition is used to assemble these terms with unknown coefficients that can be solved as a set of linear equations using the input-response data evaluated at the design points. The reduction is trivially done by eliminating those terms with very small coefficients. In this context, the main contribution of the present work lies not only in the use of moment quadrature for numerical integration, but in leveraging moment quadrature rules as a deterministic and distribution-consistent design point generator for surrogate model construction. The resulting MQDPs form a unified computational basis that simultaneously supports surrogate fitting, statistical moment evaluation, and global sensitivity analysis. In contrast to Gaussian quadrature–based PCE methods, where quadrature rules are primarily used for coefficient projection, the proposed framework treats MQDPs as general-purpose design points for surrogate modeling. Furthermore, by incorporating sparse-grid constructions, the methodology is extended to high-dimensional problems while maintaining a fully deterministic, non-adaptive formulation. Moreover, a streamlined construction and reduction procedure is formulated for any generic computable model with finite-dimensional random inputs and scalar quantities of interest, allowing the method to be adopted in general scenarios where special tuning and treatment is less practical.

The remainder of this paper is organized as follows. Section 2 introduces the mathematical foundation of the moment quadrature method and presents the deterministic design point generation scheme. Section 3 formulates the MQDP-based surrogate construction methodology, including both full-tensor and sparse implementations. Section 4 provides benchmark and engineering application examples to demonstrate the accuracy, efficiency, and scalability of the proposed framework. Finally, section 5 concludes the study and discusses potential directions for future work.

2. Deterministic design point generation via the moment quadrature method

In this study, the design points are directly generated as the quadrature nodes by the moment quadrature method. The rationality of doing so can be explained by the following reasons. As the quadrature nodes and weights are derived from the given PDF and are Gaussian, an n-node quadrature rule has a maximum precision of 2 n–1 in terms of the polynomial degree of the integrand. Hence when the subsequent surrogate basis functions are nothing more than polynomial terms, the resulting surrogate model can offer the maximum degree of accuracy in terms of the mean, variance, and even higher-order statistics such as skewness and kurtosis. The maximum degree of accuracy is a direct property of the powerful Gauss quadrature. For all possible n-node quadrature rules, such as Newton–Cotes, Clenshaw–Curtis, Gauss–Lobatto, etc., the Gauss quadrature rule yields the highest polynomial exactness. Another important practical fact is that not all design variables are uniformly distributed in a range. This makes the random generation of sample points using LHS arbitrary for distributions other than uniform ones. Hence, the goal here is to devise a design point generation scheme based on a deterministic procedure that is suitable for all kinds of PDFs, not only uniform ones. To begin with, the moment quadrature method is briefly outlined using the one-dimensional case. After that, the multi-dimensional case can naturally be obtained as a full tensor product possibly followed by a sparse rule to reduce the final size of the design points. Eventually, the sub-procedures are packed into an algorithmic procedure for general-purpose implementation.

It is important to distinguish the proposed MQDP-based surrogate modeling framework from classical Gaussian quadrature–based PCE methods. In conventional PCE, Gaussian quadrature rules are primarily employed to project model responses onto polynomial bases for coefficient estimation. In contrast, MQDPs are constructed as deterministic design points for surrogate fitting, independent of an explicit polynomial expansion of the response. The same MQDP set supports surrogate construction, numerical integration for moment estimation, and Sobol' sensitivity analysis, resulting in a unified and reusable computational pipeline. While the ability to generate quadrature nodes for arbitrary probability distributions is an important advantage, the usefulness of the proposed method lies in its deterministic design-point philosophy and unified surrogate, UQ, and sensitivity workflow rather than distribution universality alone.

2.1. The moment quadrature method

The moment quadrature method recently developed has shown great potential for uncertainty modeling and quantification [37–40], particularly for real-world problems involving computation-demanding models for which the use of the Monte Carlo method and its variants may be restricted. The basic idea of the moment quadrature method is to derive a set of Gauss quadrature rules from the Hankel matrix of statistical moments associated with the PDF of a random variable. When only a finite number of data samples instead of a PDF, are available, the empirical moments can be used instead of the theoretical statistical moments in principle.

Consider a probability space denoted by the triplet $(\Omega, \mathbb{A}, P)$ where Ω, $\mathbb{A}$, and P are an outcome space, a σ-algebra, and a probability measure, respectively. Hence, $P(A)$ defines the

occurrence probability of an event $A \in \mathbb{A}$. Denote X as a continuous random variable and x as a random instance of X. The cumulative distribution function (CDF) of x is written as

$$F(x) = P(X \leqslant x). \tag{1}$$

The PDF of x is

$$\mathbb{P}(x) = \frac{\mathrm{d}F(x)}{\mathrm{d}x}, \tag{2}$$

when the derivative exists. The kth raw statistical moment μ_k can be expressed as

$$\mu_k = \mathbb{E}_{\mathbb{P}(x)}\left[x^k\right] = \int_{x\in\Omega} x^k p(x)\,\mathrm{d}x, \tag{3}$$

where $k = 0, 1, 2, \ldots$ and $\mathbb{E}_{\mathbb{P}(x)}[\cdot]$ denotes the mathematical expectation under the PDF of $p(x)$. The zeroth moment is the integral of a PDF and is 1. Using the first $2k$th raw moments, the symmetric Hankel matrix of moments can be formed as

$$H_k = \begin{bmatrix} \mu_0 & \mu_1 & \mu_2 & \cdots & \mu_k \\ \mu_1 & \mu_2 & \mu_3 & \cdots & \mu_{k+1} \\ \mu_2 & \mu_3 & \mu_4 & \cdots & \mu_{k+2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mu_k & \mu_{k+1} & \mu_{k+2} & \cdots & \mu_{2k} \end{bmatrix}. \tag{4}$$

The Hankel matrix of moments is strictly positive since the entries are moments of a PDF; hence the matrix can alternatively be expressed as

$$H_k = R^{\mathrm{T}} R, \tag{5}$$

where R is the upper Cholesky triangular matrix, and the superscript T represents the transpose. Denote r_{ij}, $i \leqslant j$, $i,j = 1, 2, \ldots, k+1$ as the entries of R such that

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} & \cdots & r_{1,k+1} \\ & r_{22} & r_{23} & \cdots & r_{2,k+1} \\ & & r_{33} & \cdots & r_{3,k+1} \\ & & & \ddots & \vdots \\ & & & & r_{k+1,k+1} \end{bmatrix}. \tag{6}$$

The Mysovskih theorem [41] states that the entries of R correspond to an orthogonal polynomial system $\phi_1(x), \phi_2(x), \ldots, \phi_k(x)$ of order k via the following three-term recurrence equation

$$x\phi_{j-1}(x) = \beta_{j-1}\phi_{j-2}(x) + \alpha_j\phi_{j-1}(x) + \beta_j\phi_j(x), \tag{7}$$

where the coefficients are

$$\begin{cases} \beta_j = \frac{r_{j+1,j+1}}{r_{jj}} \\ \alpha_j = \frac{r_{j-1,j}}{r_{j-1,j-1}} + \frac{r_{j,j+1}}{r_{jj}} \end{cases}. \tag{8}$$

Note that $\phi_{-1}(x) \equiv 0$, $\phi_0(x) \equiv 1$, $r_{00} = 1$, $r_{01} = 0$, and the remaining terms are determined by the entries of R. Using equation (7), the orthogonal polynomial system can now be compactly expressed as

$$x\begin{bmatrix} \phi_0(x) \\ \phi_1(x) \\ \phi_2(x) \\ \vdots \\ \phi_{k-1}(x) \end{bmatrix} = \begin{bmatrix} \alpha_1 & \beta_1 & 0 & \cdots & 0 \\ \beta_1 & \alpha_2 & \beta_2 & \cdots & 0 \\ 0 & \beta_2 & \alpha_3 & \beta_3 & \vdots \\ \vdots & \vdots & \vdots & \ddots & \beta_{k-1} \\ 0 & 0 & \cdots & \beta_{k-1} & \alpha_k \end{bmatrix} \times \begin{bmatrix} \phi_0(x) \\ \phi_1(x) \\ \phi_2(x) \\ \vdots \\ \phi_{k-1}(x) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ \beta_k\phi_k(x) \end{bmatrix}. \tag{9}$$

Denote the Jacobi matrix in equation (9) as J, and

$$J = \begin{bmatrix} \alpha_1 & \beta_1 & 0 & \cdots & 0 \\ \beta_1 & \alpha_2 & \beta_2 & \cdots & 0 \\ 0 & \beta_2 & \alpha_3 & \beta_3 & \vdots \\ \vdots & \vdots & \vdots & \ddots & \beta_{k-1} \\ 0 & 0 & \cdots & \beta_{k-1} & \alpha_k \end{bmatrix}. \tag{10}$$

Denote the orthonormal eigenvectors and eigenvalues of J as $\mathbb{V}$ and $\mathbb{D}$, respectively, and they can be solved via the solver function eig$(\cdot)$ such that

$$[\mathbb{V}, \mathbb{D}] = \mathrm{eig}(J). \tag{11}$$

It is shown that the eigenvalue of the Jacobi matrix J and the square of the component of the corresponding eigenvector form a Gauss quadrature rule under the weight $p(x)$ [42]. The nodes and weights of the rule are (ξ_j, w_j), $j = 1, \ldots, k$. These nodes and weights are associated with $[\mathbb{V}, \mathbb{D}]$ through

$$\begin{cases} \xi_j = \mathbb{V}_j \\ w_j = \mathbb{D}_{j1}^2 \end{cases}, \tag{12}$$

where $\mathbb{V}_j$ is the jth eigenvalue, and $\mathbb{D}_{j1}$ is the first component of the jth eigenvector. It should be emphasized that the size of the Hankel matrix H_k is determined solely by the quadrature order in each individual dimension and is independent of the overall stochastic dimension of the problem. Therefore, high-dimensional surrogate construction does not enlarge the matrix H_k. Since the Jacobi matrix in equation (10) is symmetric and tridiagonal, its eigen-decomposition is numerically stable. In practical surrogate modeling, only moderate quadrature orders are required, and numerical instability associated with the Cholesky decomposition of very high-order moment matrices is usually not encountered. When very high-order moment matrices are indeed needed, recurrence formula developed in [43] can be adopted to directly obtain the elements (α_i, β_i) in equation (10), avoiding the Cholesky decomposition of H_k.

Given a general integrand $\psi(x)$, the integral under the PDF $p(x)$, which is the mathematical expectation of $\psi(x)$, denoted as $\mathbb{E}_p[\psi(x)]$, can be evaluated as

$$\mathbb{E}_p[\psi(x)] = \int_{\Omega} \psi(x) p(x)\,\mathrm{d}x \approx \sum_{j=1}^{k} \psi(\xi_j) w_j. \tag{13}$$

As a k-node Gauss quadrature rule is exact for a polynomial integrand up to order $2k$–1; hence the approximately equal sign is replaced by an equal sign for any polynomial integrand $\psi(x)$ up to order $2k$–1. It is natural to see that replacing $\psi(x)$ by $\psi^m(x)$ in equation (13) yields

$$\mathbb{E}_p[\psi^m(x)] = \int_{\Omega} \psi^m(x) p(x)\,\mathrm{d}x \approx \sum_{j=1}^{k} \psi^m(\xi_j) w_j, \tag{14}$$

which is the mth raw statistical moment of $\psi(x)$, and $m = 1,\ldots,2k-1$. Hence, the statistical moments of a function under continuous PDFs can directly be obtained with a total number of k model runs in the univariate case. Extending the quadrature rule to the multi-dimensional case with uncorrelated variables can be done via the tensor product of the univariate rules of all participating variables. For linearly correlated multi-dimensional variables, i.e. when the correlation is expressed by a correlation matrix or a variance-covariance matrix, the nodes can readily be obtained by pre-multiplying the node vector with the Cholesky decomposition triangular matrix of the correlation matrix or the variance-covariance matrix.

2.2. MQDPs in one-dimensional variable space

The implication of equations (13) and (14) is that when the nodes in equation (12), $\xi_j, j = 1,\ldots,k$, and the corresponding true model responses, $\psi(\xi_j)$, are used to build a polynomial surrogate model, i.e.

$$M(x) = c_0 + c_1 x + c_2 x^2 + \ldots + c_{k-1} x^{k-1}, \tag{15}$$

the mean and variance of $\mathcal{M}(x)$ can exactly be obtained. This is shown by the fact that the variance of $\mathcal{M}(x)$ has the highest power of $2(k-1)$ and is less than $2k-1$. Moreover, the coefficients can readily be found by solving the following set of linear equations

$$\begin{bmatrix} \mathcal{M}(\xi_1) \\ \mathcal{M}(\xi_2) \\ \vdots \\ \mathcal{M}(\xi_k) \end{bmatrix} = \begin{bmatrix} 1 & \xi_1 & \xi_1^2 & \cdots & \xi_1^{k-1} \\ 1 & \xi_2 & \xi_2^2 & \cdots & \xi_2^{k-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \xi_{k-1} & \xi_{k-1}^2 & \cdots & \xi_{k-1}^{k-1} \end{bmatrix} \cdot \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_{k-1} \end{bmatrix}. \tag{16}$$

To identify the unknown parameters $\mathbf{c} = [c_0, c_1, \cdots, c_{k-1}]^{\mathrm{T}}$, it is no more than equating the right-hand side of equation (16) to the true model response vector whose components are the true model response evaluated at $\xi_1,\ldots,\xi_k$. In matrix form

$$c = [\boldsymbol{\xi}]^{-1} \mathrm{y}(\boldsymbol{\xi}), \tag{17}$$

where $\mathbf{c}$ denote the column vector of $[c_0, c_1, \ldots, c_{k-1}]^{\mathrm{T}}$, $\mathrm{y}(\boldsymbol{\xi})$ is the model response vector, and $[\boldsymbol{\xi}]$ is the matrix of the first term on the right-hand-side of equation (16). With equation (17), the polynomial surrogate model prediction writes,

$$\begin{aligned} M(x) &= \left[1, x, x^2, \ldots, x^{k-1}\right] \\ c &= \left[1, x, x^2, \ldots, x^{k-1}\right] \left([\boldsymbol{\xi}]^{-1} \mathrm{y}(\boldsymbol{\xi})\right). \end{aligned} \tag{18}$$

In this one-dimensional case, the k-element design point set is the nodes resulting from the k-node moment quadrature rule. For convenience, denote the design points as a whole as a k-element column vector

$$\mathrm{DP}_x(k) := [\xi_1, \xi_2, \ldots, \xi_k]^{\mathrm{T}}. \tag{19}$$

2.3. MQDPs in multi-dimensional variable space

Extending to a multi-dimensional variable space is nothing more than extending the quadrature rule from the one-dimensional case to the multi-dimensional case while still treating these quadrature nodes as design points. The extension is made via tensor product as follows. First, for each of the random variables with its PDF, generate the corresponding design points. The number of design points for individual random variables can be different depending on prescribed constraints and/or one's choice. Next, assemble the set of generated one-dimensional design points via a tensor product, and obtain the final design points of the multi-dimensional variable space. Denote the d-dimensional random vector as $\mathbf{x} = [x_1, x_2, \ldots, x_d]$, and for the ith random variable, $i = 1,\ldots,d$, there are k_i univariate design points associated with the random variable that are generated according to the procedure in section 2.2. Consequently, the final design points in the d-dimensional variable space are

$$\mathrm{DP}_{\mathbf{x}}(\kappa) := \mathrm{DP}_{x_1}(k_1) \otimes \mathrm{DP}_{x_2}(k_2) \otimes \cdots \otimes \mathrm{DP}_{x_d}(k_d), \tag{20}$$

where the operator $\otimes$ is the tensor product applied to two vectors, e.g. $[1,2]^{\mathrm{T}} \otimes [3,4]^{\mathrm{T}} = \begin{bmatrix} 1 & 1 & 2 & 2 \\ 3 & 4 & 3 & 4 \end{bmatrix}^{\mathrm{T}}$, and $\kappa = \prod_{i=1}^{d} k_i$. It is evident that the design points of equation (20) form a κ-by-d matrix.

For linearly correlated random variables, i.e. the correlation can be expressed by a correlation matrix and/or a variance-covariance matrix, the final design points are obtained by pre-multiplying the triangular matrix resulting from the Cholesky decomposition of the correlation matrix. Denote the correlation matrix of a d-dimensional random vector as $\boldsymbol{\rho}$, and L_ρ as the lower triangular matrix of the Cholesky decomposition of the matrix $\boldsymbol{\rho}$ such that

$$\rho = L_\rho L_\rho^{\mathrm{T}} = \begin{bmatrix} 1 & \rho_{12} & \cdots & \rho_{1d} \\ \rho_{21} & 1 & \cdots & \rho_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ \rho_{d1} & \rho_{d2} & \cdots & 1 \end{bmatrix}, \tag{21}$$

where $\rho_{ij} = \rho_{ji}$ for $i,j = 1,\ldots,d$ and $i \neq j$. The design points for the d-dimensional correlated random vector can readily be obtained as

$$\mathrm{DP}_{\mathbf{x}}(\kappa,\rho) := [\mathrm{DP}_{x_1}(k_1) \otimes \mathrm{DP}_{x_2}(k_2) \otimes \cdots \otimes \mathrm{DP}_{x_d}(k_d)] \cdot L_{\rho}^{\mathrm{T}}, \tag{22}$$

which remains a κ-by-d matrix, and each row of the matrix is a d-dimensional design point in the variable space. These design points generated by the moment quadrature method are referred to as MQDPs hereafter.

3. Surrogate model construction methodology

The construction of the surrogate model for the one-dimensional case is straightforward as shown in equation (15), and the discussion here is focused on the multi-dimensional case instead.

3.1. Standard procedure for constructing MQDP polynomial surrogate

Consider a set of MQDPs with κ points at $\mathbf{x}_i$, $i = 1,2,\ldots,\kappa$, where $\mathbf{x}^T = [x_1]$ for the one-dimensional case, $\mathbf{x}^T = [x_1,x_2]$ for the two-dimensional case, and $\mathbf{x}^T = [x_1,x_2,x_3]$ for the three-dimensional case. The same notation can be extended to higher dimensions, e.g. for the d-dimensional case, $\mathbf{x}^T = [x_1,x_2,\ldots,x_d]$ for $d \in \mathbb{N}$. For one-dimensional cases, a basis of a complete p-order polynomial is the set

$$\mathbb{P}(\mathbf{x}) = \left\{1, x_1, x_1^2, x_1^3, \ldots, x_1^p\right\}. \tag{23}$$

In this case, the number of MQDPs required to identify the polynomial coefficients is $p+1$. Evidently, extending to d-dimensional cases can be done by enumerating all the terms through the tensor product grid of monomials schematically shown in figure 1. Figure 1(a) shows the monomial grid resulting from the tensor product of the two one-dimensional bases $\left\{1,x_1,x_1^2,\ldots,x_1^5\right\}$ and $\left\{1,x_2,x_2^2,\ldots,x_2^5\right\}$ in the two-dimensional space. Similarly, the tensor product of the bases $\left\{1,x_1,x_1^2\right\}$, $\left\{1,x_2,x_2^2\right\}$, and $\left\{1,x_3,x_3^2\right\}$ generates the monomial grid in figure 1(b) in the three-dimensional space. It is possible to generate a grid with different maximum orders in each dimension. For example, $\left\{1,x_1,x_1^2\right\} \otimes \left\{1,x_2,x_2^2,x_2^3\right\} \otimes \left\{1,x_3\right\}$ generates a tensor grid with 24 nodes, and in this case $\mathrm{p}(x)$ forms the set of 24 monomial terms. Moreover, the polynomial terms in the three dimensions are 3, 4, and 2, respectively, and the corresponding numbers of MQDPs are 3, 4, and 2, respectively.

The nodal monomial terms thus form the basis of the polynomial surrogate model. For the one shown in figure 1(a), the number of the basis terms is 25, and the maximum monomial order is 10, which is the summation of the maximum orders in each dimension. The resulting basis is the set

$$\mathbb{P}(\mathbf{x}) = p(x_1,x_2) = \left\{1, x_1, x_2, x_1x_2, x_1^2, x_2^2, \ldots, x_1^5x_2^5\right\}. \tag{24}$$

For the grid shown in figure 1(b), the maximum order is 6 and the basis is the set of nodal monomials

$$\begin{aligned} p(\mathbf{x}) &= p(x_1,x_2,x_3) \\ &= \left\{1, x_1, x_2, x_3, x_1x_2, x_2x_3, x_3x_1, x_1^2, x_2^2, x_3^2, \ldots, x_1^2x_2^2x_3^2\right\}. \end{aligned} \tag{25}$$

Sum up all the basis monomial terms with their coefficients to obtain the polynomial surrogate model

$$M(\mathbf{x}) = p(\mathbf{x})\,\boldsymbol{\alpha}^{\mathrm{T}}, \tag{26}$$

where $\boldsymbol{\alpha} = [\alpha_1,\alpha_2,\ldots,\alpha_\kappa]$ are the vector of coefficients to be identified, and κ is the number of monomials in $p(\mathbf{x})$. It is noted that the number of coefficients is identical to the number of MQDPs; therefore, the coefficients can be resolved by solving the set of linear equations

$$\begin{bmatrix} \mathrm{y}(\boldsymbol{\xi}_1) \\ \mathrm{y}(\boldsymbol{\xi}_2) \\ \mathrm{y}(\boldsymbol{\xi}_3) \\ \vdots \\ \mathrm{y}(\boldsymbol{\xi}_\kappa) \end{bmatrix} = \begin{bmatrix} \mathbb{P}(\mathbf{x}_1|\boldsymbol{\xi}_1) \\ \mathbb{P}(\mathbf{x}_2|\boldsymbol{\xi}_2) \\ \mathbb{P}(\mathbf{x}_3|\boldsymbol{\xi}_3) \\ \vdots \\ \mathbb{P}(\mathbf{x}_\kappa|\boldsymbol{\xi}_\kappa) \end{bmatrix} \cdot \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \vdots \\ \alpha_\kappa \end{bmatrix} \tag{27}$$

to have

$$\boldsymbol{\alpha}^{\mathrm{T}} = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \vdots \\ \alpha_\kappa \end{bmatrix} = \begin{bmatrix} \mathbb{P}(\mathbf{x}_1|\boldsymbol{\xi}_1) \\ \mathbb{P}(\mathbf{x}_2|\boldsymbol{\xi}_2) \\ \mathbb{P}(\mathbf{x}_3|\boldsymbol{\xi}_3) \\ \vdots \\ \mathbb{P}(\mathbf{x}_\kappa|\boldsymbol{\xi}_\kappa) \end{bmatrix}^{-1} \cdot \begin{bmatrix} \mathrm{y}(\boldsymbol{\xi}_1) \\ \mathrm{y}(\boldsymbol{\xi}_2) \\ \mathrm{y}(\boldsymbol{\xi}_3) \\ \vdots \\ \mathrm{y}(\boldsymbol{\xi}_\kappa) \end{bmatrix}. \tag{28}$$

In equations (27) and (28), the term $p(\mathbf{x}_i|\boldsymbol{\xi}_i)$, $i = 1,\cdots,\kappa$ denotes the monomial term vector evaluated at the ith design point $\boldsymbol{\xi}_i$, and $\mathrm{y}(\boldsymbol{\xi}_i)$, $i = 1,\cdots,\kappa$ denotes the true model evaluated at the ith design point. For example, given the monomial term vector in equation (25), $p(\mathbf{x}_1|\boldsymbol{\xi}_1) = \left\{1,\xi_{11},\xi_{12},\xi_{13},\xi_{11}\xi_{12},\xi_{12}\xi_{13},\xi_{13}\xi_{11},\cdots,\xi_{11}^2\xi_{12}^2\xi_{13}^2\right\}$, and ξ_{11}, ξ_{12}, and ξ_{13} are the values of the first DP for the three variables.

Note that the first term on the right-hand-side of equation (27), i.e. $[\mathbb{P}(\mathbf{x}_1),p(\mathbf{x}_2),p(\mathbf{x}_3),\cdots,p(\mathbf{x}_\kappa)]^T$, is a square matrix, so the inversion exists. The final MQDP surrogate model for predicting the response at input $\mathbf{x}$ is now complete as

$$M(\mathbf{x}) = p(\mathbf{x}) \cdot \left(\begin{bmatrix} \mathbb{P}(\mathbf{x}_1|\boldsymbol{\xi}_1) \\ \mathbb{P}(\mathbf{x}_2|\boldsymbol{\xi}_2) \\ \mathbb{P}(\mathbf{x}_3|\boldsymbol{\xi}_3) \\ \vdots \\ \mathbb{P}(\mathbf{x}_\kappa|\boldsymbol{\xi}_\kappa) \end{bmatrix}^{-1} \cdot \begin{bmatrix} \mathrm{y}(\boldsymbol{\xi}_1) \\ \mathrm{y}(\boldsymbol{\xi}_2) \\ \mathrm{y}(\boldsymbol{\xi}_3) \\ \vdots \\ \mathrm{y}(\boldsymbol{\xi}_\kappa) \end{bmatrix} \right). \tag{29}$$

To generate the multi-dimensional MQDPs using the MQDPs of each variable, the function 'combvec' provided in appendix A2 of [39] can be used. It is particularly suitable for

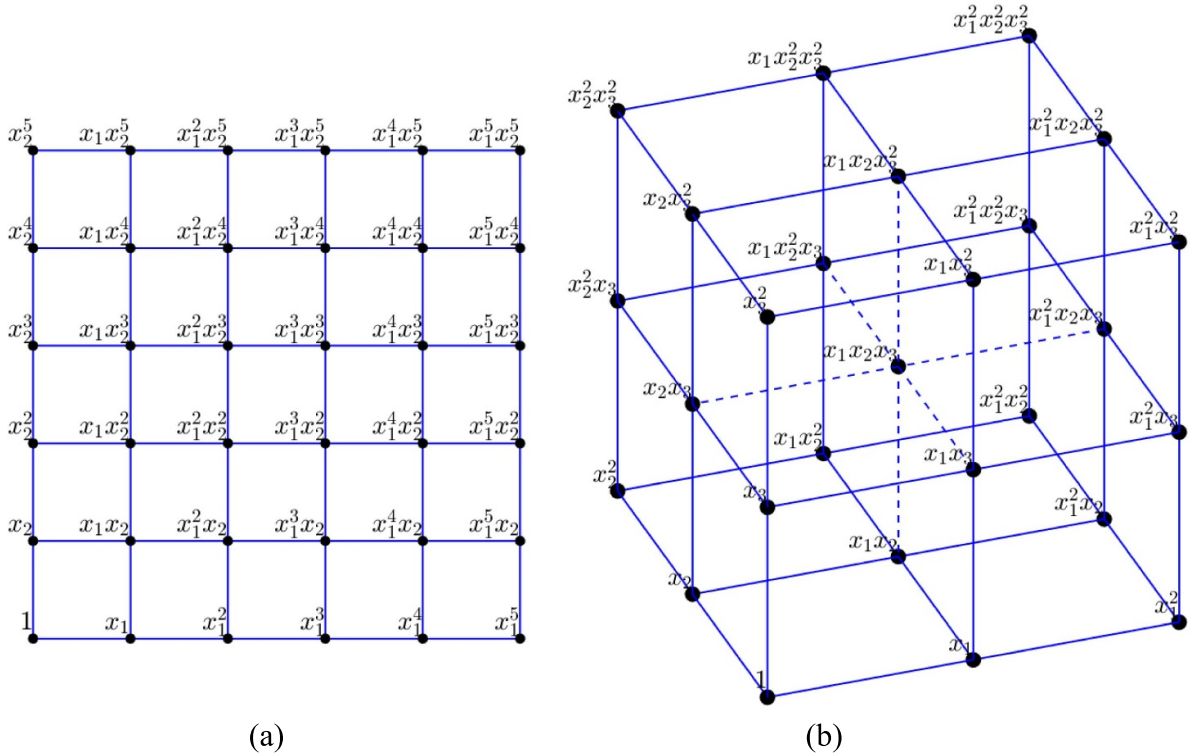


Figure 1. (a) Tensor product grid in the two-dimensional space using the bases of order 5 in each dimension, and (b) tensor product grid in the three-dimensional space using the bases of order 2 in each dimension.

high-dimensional problems with several nodes in each dimension, as in those scenarios the manual listing of the tensor product would be less efficient. More importantly, the function 'combvec' is also suitable for the generation of monomial terms in the basis vector of $p(\mathbf{x})$ with the aid of the symbolic toolbox; consequently, the whole procedure from generation of MQDPs to the generation of monomials of the polynomial surrogate and the final regression of fitting coefficients can all be done in a streamlined computational pipeline. This allows an arbitrary and computable black-box model to be surrogated without special treatment or manual adjustment of the computational codes and algorithms. The whole procedure of constructing a surrogate model using MQDPs is summarized in figure 2.

A simple example is given here to substantiate the procedure. Consider a 3-dimensional problem with $\mathbf{x} = [x_1, x_2, x_3]$. Two MQ nodes in each dimension are used, resulting in a total number of 2^3 MQDPs in the three-dimensional variable space. The distributions of the three random variables x_1, x_2, and x_3 are uniform in [0, 1], Gaussian with zero mean and unit variance, and exponential with a rate of 1, respectively. Hence the basis of the surrogate model is the tensor product of $\{1, x_1\} \otimes \{1, x_2\} \times \{1, x_3\}$, and therefore $p(\mathbf{x}) = \{1, x_1, x_2, x_3, x_1x_2, x_2x_3, x_3x_1, x_1x_2x_3\}$, i.e. a tri-linear monomial grid shown in figure 3(a). The MQ nodes for each variable in the 3-dimensional variable space are shown in figure 3(b).

In the example case, the two nodes for uniform, Gaussian, and exponential distributions are found as [0.21132, 0.78868], $[-1, 1]$, and [0.58879, 3.4142], respectively, following the calculation steps in section 2.1. The tensor product of the three sets of nodes yields the set of MQDPs

$$\mathrm{DP}_{\mathbf{x}}(8) = \begin{bmatrix} 0.2113 & 0.7887 & 0.2113 & 0.7887 & 0.2113 & 0.78868 & 0.2113 & 0.7887 \\ -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 \\ 0.5858 & 3.4142 & 0.5858 & 3.4142 & 0.5858 & 3.4142 & 0.5858 & 3.4142 \end{bmatrix}^T. \tag{30}$$

Using the MQDPs in equation (30) to evaluate the true model, and the eight model responses can readily be used to obtain the fitting coefficients. To motivate the discussion, $y(\mathbf{x}) = x_1x_2x_3$ is assumed, and the fitting coefficients are obtained as $\mathbf{c} = [0, 0, 0, 0, 0, 0, 0, 1]^T$, which reproduces the true model. Despite the simplicity of the toy problem example, one notable fact is that certain coefficients are zero, which indicates that those monomial terms with

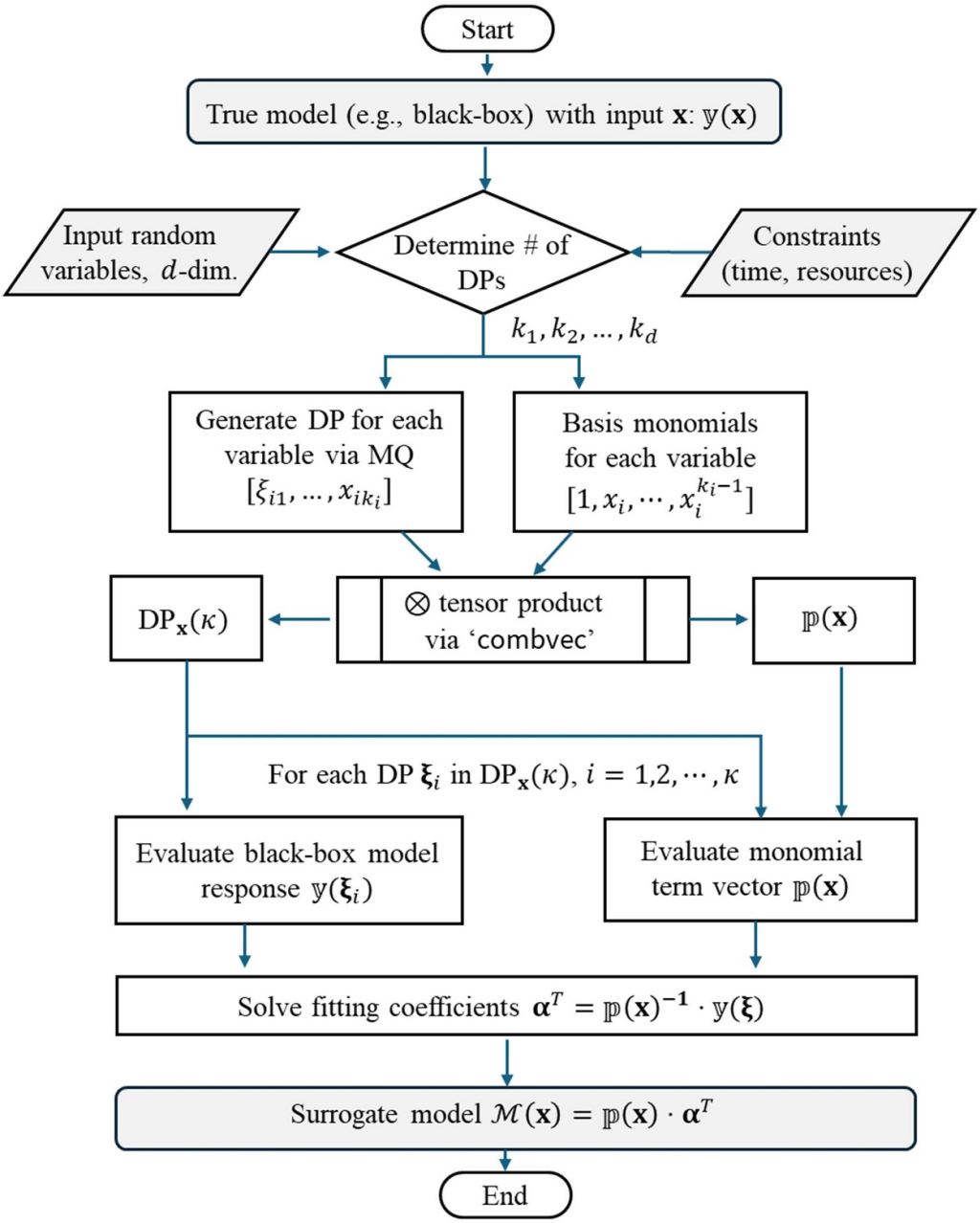


Figure 2. The standard procedure of MQDP surrogate model construction.

zero or almost zero coefficients in $p(x)$ can be eliminated. The accuracy of the surrogate model is assured by the Weierstrass approximation theorem, which states that any continuous real-valued function defined on a closed interval can be approximated as closely as desired by a polynomial function [44]. As the number of MQDPs is equal to the number of MQ nodes, which determines the polynomial exactness of the surrogate, the accuracy of the surrogate therefore increases monotonically as the number of MQ nodes increases.

3.2. Standard procedure for constructing sparse MQDP polynomial surrogate

The curse of dimensionality can limit the construction of a full-tensor surrogate model as the dimension of the problem becomes larger. The k-node one-dimensional DPs can produce k^d total number of MQDPs in the d-dimensional variable space, which corresponds to k^d monomial terms whose coefficients in turn require the inversion of a k^d-by-k^d matrix or the solution of a set of k^d linear equations when an iterative method

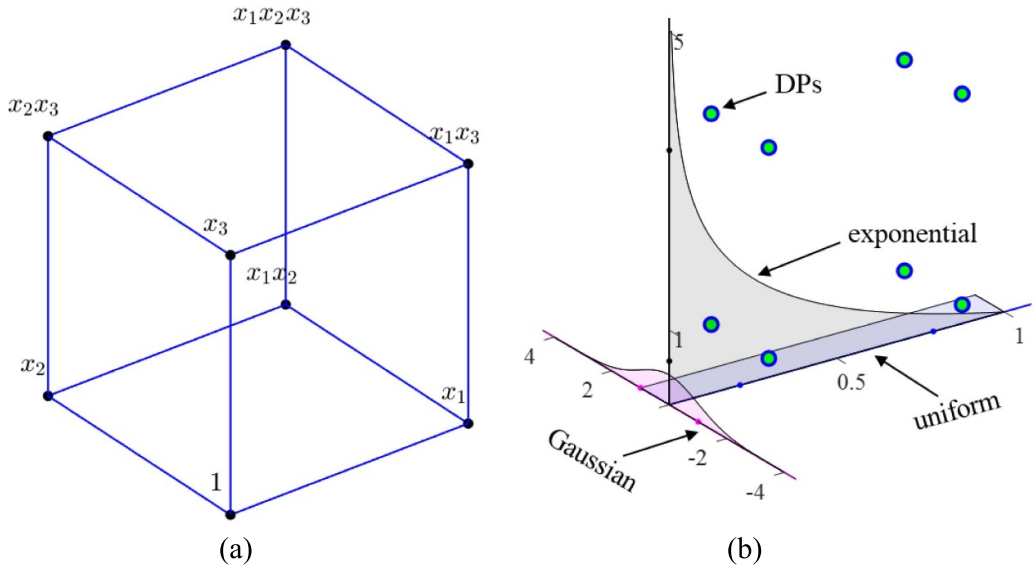


Figure 3. (a) Tensor product grid showing nodal terms as monomials, and (b) the MQ nodes for each variable.

is used. In either case, the computation can be intractable when d is large enough, e.g. $d \geqslant 20$. Hence, it is necessary to extend the standard procedure to deal with high-dimensional cases. To achieve that, the sparse moment quadrature (SMQ) method can be adopted to generate a sparse set of MQDPs. The SMQ is formulated by incorporating the Smolyak sparse rule into the standard MQ method.

The basic idea of the Smolyak rule is to omit the higher-order interaction terms in the linear functional approximating the true function. For example, the true function can be a multi-dimensional integral while the linear functional is the Gauss quadrature formula in the SMQ method. It is assumed that these omitted terms have a negligible effect in the accuracy of the linear functional, which is true for many practical problems as several studies suggest [45–47]. In other words, a truncated version of the full-tensor quadrature formula is obtained by prescribing a level parameter. The level parameter and the number of dimensions collectively determine the maximum order in the truncated set. In the context of surrogate model building, the corresponding sparse tensor product reflects a set of sparse monomial terms for building the sparse surrogate model. This is analogous to the case in which the full-tensor MQ reflects a full-tensor set of monomials in section 2.1. By reducing the full tensor set to the sparse set, the exponential scaling of the number of terms can be brought down to polynomial scaling, and to a linear scaling at most. The level parameter also influences the polynomial exactness of the linear functional; hence it is possible to increase the level parameter to enhance the accuracy of the approximation.

The detailed method for generating SMQ nodes and weights can be found in [39], and the generation of the sparse monomial terms is of concern here. Denote $\mathcal{F}^{(i)}$ $i = 1, \ldots, d$ as one-dimensional linear operators, e.g. the expectation operator $\mathbb{E}_p$ in equation (13). For the d-dimensional linear operator, the tensor product of all possible combinations is written as

$$\mathcal{F}^{(d)} = \mathcal{F}^{(1)} \otimes \mathcal{F}^{(2)} \otimes \cdots \otimes \mathcal{F}^{(d)} = \bigotimes_{i=1}^{d} \mathcal{F}^{(i)}. \tag{31}$$

In practice, the linear functional may not have a closed-form expression, and an approximation linear functional, namely, the k_i-node ($k_i \in \mathbb{N}$) moment quadrature, is used to approximate $\mathcal{F}^{(i)}$. Denote the approximation functional as $\mathcal{F}_{k_i}^{(i)}$

$$\mathcal{F}_{k_i}^{(i)} f = \sum_{j=1}^{k_i} w_j f(\xi_j), \tag{32}$$

and

$$\lim_{k_i \to \infty} \| \mathcal{F}^{(i)} f - \mathcal{F}_{k_i}^{(i)} f \| = 0. \tag{33}$$

The term $\mathcal{F}_{k_i}^{(i)}$ can be expressed as a telescoping sum,

$$\begin{aligned} \mathcal{F}_{k_i}^{(i)} = \mathcal{F}_1^{(i)} + \left(\mathcal{F}_2^{(i)} - \mathcal{F}_1^{(i)} \right) + \left(\mathcal{F}_3^{(i)} - \mathcal{F}_2^{(i)} \right) \\ + \ldots + \left(\mathcal{F}_{k_i}^{(i)} - \mathcal{F}_{k_i-1}^{(i)} \right). \end{aligned} \tag{34}$$

Using equations (34) and (33), $\mathcal{F}^{(i)}$ is written as

$$\mathcal{F}^{(i)} = \sum_{j=1}^{\infty} \left(\mathcal{F}_j^{(i)} - \mathcal{F}_{j-1}^{(i)} \right), \tag{35}$$

where $\mathcal{F}_0^{(i)} := 0$. Let $\Delta_j^{(i)} = \mathcal{F}_j^{(i)} - \mathcal{F}_{j-1}^{(i)}$, equation (35) becomes

$$\mathcal{F}^{(i)} = \sum_{j=1}^{\infty} \Delta_j^{(i)}, \tag{36}$$

with $\Delta_1^{(i)} := \mathcal{F}_1^{(i)}$. For the d-dimensional case, substituting equation (36) into equation (31) gives

$$\mathcal{F}^{(d)} = f^{(1)} \otimes \cdots \otimes f^{(d)} = \left(\sum_{j_1=1}^{\infty} \Delta_{j_1}^{(1)} \right) \otimes \cdots \otimes \left(\sum_{j_d=1}^{\infty} \Delta_{j_d}^{(d)} \right),$$
$$= \sum_{j_1=1}^{k_1 \to \infty} \cdots \sum_{j_1=1}^{k_d \to \infty} \bigotimes_{i=1}^{d} \Delta_{j_i}^{(i)} = \sum_{\|k\|_1=1}^{\infty} \bigotimes_{i=1}^{d} \Delta_{j_i}^{(i)} \tag{37}$$

where $\mathbb{k} = (j_1, j_2, \ldots, j_d) \in \mathbb{N}^d$ is the multi-index for the nodes in each dimension, $\|\mathbb{k}\|_1 := \sum_{i=1}^{d} |j_i|$ is the ℓ^1 norm, and $j_i = \{1, 2, \ldots, k_i\}$ for $i = 1, \ldots, d$. It is noted that equation (37) is exact but has an infinite number of terms, and it is evident that $\Delta_{k_i}^{(i)} \to 0$ as $k_i \to \infty$; hence it is possible to omit terms with little influence on the accuracy. It can be highly nontrivial in practice to identify important terms apriori; therefore, it is intuitive to truncate the full set by introducing a level parameter L such that the terms with $\|\mathbb{k}\|_1 \leqslant L + d$ are kept. This linear truncation is equivalent to truncating along the diagonal of the full tensor grid. The truncated version of the linear functional can be deduced using the combinational method as

$$f^{(d)} \approx F_{d+L}^{(d)} = \sum_{\substack{\max\{d, L+1\} \leqslant \|k\|_1 \leqslant L+d \\ \mathbb{k} \in \mathbb{N}^d, k \geqslant 1}} (-1)^{L+d-\|k\|_1}$$
$$\times \binom{d-1}{L+d-\|k\|_1} \bigotimes_{i=1}^{d} f_{j_i}^{(i)} \tag{38}$$

where $\mathbb{k} \geqslant 1$ represents element-wise larger than or equal to, and $\mathcal{F}_{j_i}^{(i)}$ is the ith dimensional univariate linear functional, i.e. j_i-node quadrature operator.

Figure 4 illustrates the truncation using a three-dimensional problem where the finite numbers of nodes in the two dimensions are $k_1 = k_2 = k_3 = 3$. The full tensor product of the nodes indexed by $(1, \ldots, k_1)$, $(1, \ldots, k_2)$, and $(1, \ldots, k_3)$ is

$$\{(j_1, j_2, j_3)\}_{\text{full}} = (1,2,3) \otimes (1,2,3) \otimes (1,2,3)$$
$$= \{(1,1,1), (2,1,1), (3,1,1), \cdots, (1,3,3), (2,3,3), (3,3,3)\}^T, \tag{39}$$

where each element of (j_1, j_2, j_3) in the set indexes the j_1th node, j_2th node, and j_3th node in the first, second, and third dimensions, respectively. The sum of an element in the multi-index set, i.e. $\|K\|_1 = j_1 + j_2 + j_3$, determines the order of the monomial term corresponding to (j_1, j_2, j_3). For example, $(j_1, j_2, j_3) = (1, 2, 2)$ corresponds to $x_1^{j_1-1} x_2^{j_2-1} x_3^{j_3-1} = x_2 x_3$. If the higher-order terms, e.g. $\|K\|_1 > 5$ can be omitted, the full multi-index set can be reduced to the sparse set

$$\{(j_1, j_2, j_3)\}_{\text{sparse}}$$
$$= \left\{ \begin{array}{c} (1,1,1), (2,1,1), (1,2,1), (1,1,2), (3,1,1), \\ (2,2,1), (1,3,1), (2,1,2), (1,2,2), (1,1,3) \end{array} \right\}^T. \tag{40}$$

Recall that the order of polynomial for a variate is $k - 1$ where k is the number of MQDPs; therefore, the sparse set in equation (40) corresponds to the monomial set

$$\mathbb{P}(\mathbf{x}) = \left\{1,\ x_1, x_2, x_3, x_1^2, x_1x_2, x_2^2, x_1x_3, x_2x_3, x_3^2\right\}. \tag{41}$$

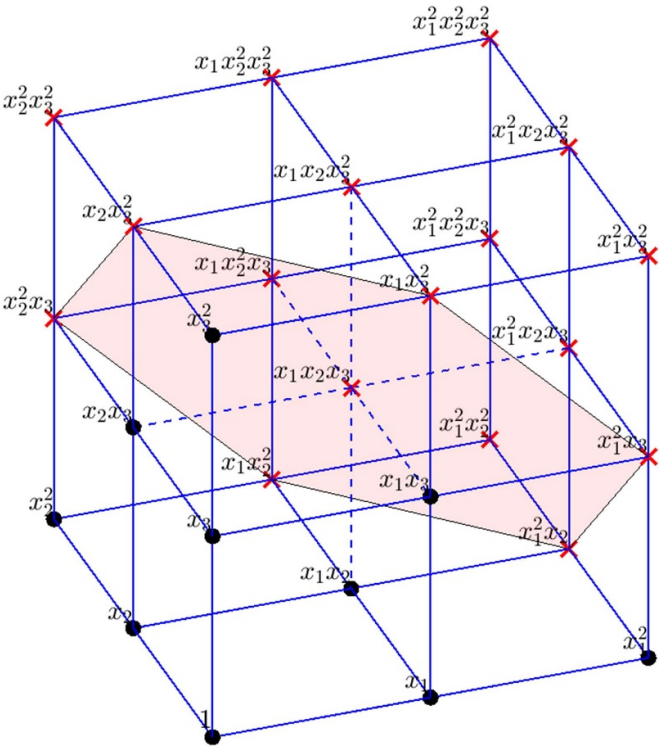


Figure 4. Terms eliminated (cross marker) and kept (solid marker) in the 3-dimensional case with the maximum retained order 2. The plane denotes the separation of the monomial tensor grid into two subdomains for the kept and eliminated terms.

The monomial terms in the set are shown as solid markers in figure 4 and the cross markers represent the eliminated monomial terms. It is readily seen that the monomials in $p(\mathbf{x})$ can be obtained via the following expression

$$m([x_1, x_2, x_3], (j_1, j_2, j_3)] = x_1^{j_1-1} x_2^{j_2-1} x_3^{j_3-1}. \tag{42}$$

More generally, for a d-dimensional case and recall and $\mathbb{k} = (j_1, j_2, \ldots, j_d) \in \mathbb{N}^d$,

$$m(\mathbf{x}, \mathbb{k}) = \prod_{i=1}^{d} x_i^{j_i-1}, \tag{43}$$

and all monomial terms can be enumerated by looping through all possible multi-indices $\mathbb{k}$ such that $\max\{d, L+1\} \leqslant \|K\|_1 \leqslant L + d$ as required by the summation operation in equation (38). For example, equation (41) can be obtained using equation (43) with each multi-index in equation (40).

In the full-tensor scenario in section 3.1, the number of DPs and the number of fitting coefficients are identical; however, this is not the case for the sparse version. In the sparse version. The number of DPs is the sum of the element-wise products

of each multi-index in the set of all possible $\mathbb{k}$, i.e.

$$\kappa = \sum_{\substack{\max\{d, L+1\} \leqslant \|K\|_1 \leqslant L+d \\ \mathrm{k} \in \mathbb{N}^d, \mathrm{k} \geqslant 1}} \prod_{i=1}^{d} j_i. \tag{44}$$

Hence the number of sparse MQDPs is always larger than the number of fitting coefficients for the sparse version of $p(\mathbf{x})$. For the multi-index set in equation (40), the total number of DPs is 28 while the total number of fitting coefficients is 10. The least squares method can be used identify the fitting coefficients using the model response evaluated at the DPs and the monomial terms such as

$$\widehat{\boldsymbol{\alpha}} = \mathrm{argmin}_{\boldsymbol{\alpha}} \left[e(\boldsymbol{\alpha})^T \cdot e(\boldsymbol{\alpha}) \right], \tag{45}$$

where e is the model residual vector

$$e(\boldsymbol{\alpha}) = \begin{bmatrix} \mathrm{y}(\boldsymbol{\xi}_1) \\ \mathrm{y}(\boldsymbol{\xi}_2) \\ \mathrm{y}(\boldsymbol{\xi}_3) \\ \vdots \\ \mathrm{y}(\boldsymbol{\xi}_\kappa) \end{bmatrix} - \begin{bmatrix} \mathbb{P}(\mathbf{x}_1|\boldsymbol{\xi}_1) \\ \mathbb{P}(\mathbf{x}_2|\boldsymbol{\xi}_2) \\ \mathbb{P}(\mathbf{x}_3|\boldsymbol{\xi}_3) \\ \vdots \\ \mathbb{P}(\mathbf{x}_\kappa|\boldsymbol{\xi}_\kappa) \end{bmatrix} \cdot \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \vdots \\ \alpha_\kappa \end{bmatrix}. \tag{46}$$

The solution to equation (45) writes

$$\widehat{\boldsymbol{\alpha}}^T = \left(\begin{bmatrix} \mathbb{P}(\mathbf{x}_1|\boldsymbol{\xi}_1) \\ \mathbb{P}(\mathbf{x}_2|\boldsymbol{\xi}_2) \\ \mathbb{P}(\mathbf{x}_3|\boldsymbol{\xi}_3) \\ \vdots \\ \mathbb{P}(\mathbf{x}_\kappa|\boldsymbol{\xi}_\kappa) \end{bmatrix}^T \begin{bmatrix} \mathbb{P}(\mathbf{x}_1|\boldsymbol{\xi}_1) \\ \mathbb{P}(\mathbf{x}_2|\boldsymbol{\xi}_2) \\ \mathbb{P}(\mathbf{x}_3|\boldsymbol{\xi}_3) \\ \vdots \\ \mathbb{P}(\mathbf{x}_\kappa|\boldsymbol{\xi}_\kappa) \end{bmatrix} \right)^{-1} \times \begin{bmatrix} \mathbb{P}(\mathbf{x}_1|\boldsymbol{\xi}_1) \\ \mathbb{P}(\mathbf{x}_2|\boldsymbol{\xi}_2) \\ \mathbb{P}(\mathbf{x}_3|\boldsymbol{\xi}_3) \\ \vdots \\ \mathbb{P}(\mathbf{x}_\kappa|\boldsymbol{\xi}_\kappa) \end{bmatrix} \begin{bmatrix} \mathrm{y}(\boldsymbol{\xi}_1) \\ \mathrm{y}(\boldsymbol{\xi}_2) \\ \mathrm{y}(\boldsymbol{\xi}_3) \\ \vdots \\ \mathrm{y}(\boldsymbol{\xi}_\kappa) \end{bmatrix}. \tag{47}$$

With the fitted coefficients, the sparse MQDP surrogate model is finally obtained as

$$M(\mathbf{x}) = p(\mathbf{x})\, \widehat{\boldsymbol{\alpha}}^T. \tag{48}$$

The overall procedure for constructing a sparse MQDP surrogate model from scratch is summarized in figure 5. The difference between the standard and sparse MQDP model constructions lies mainly in the generation of the sparse DP and the sparse monomials as well as the resolving of fitting coefficients. Moreover, the fitting residual of the sparse model always exists while for the standard model the fitting residual is absent.

One of the notable advantages of the moment quadrature surrogate modeling is that once the model responses at the design points are evaluated, the responses can directly be used to evaluate many useful quantities by mixing the weights associated with those design points, such as mean and variance for uncertainty quantification and/or Sobol' sensitivity indices for sensitivity analysis etc. By 'directly', it means that no new model runs either with the true model or with the surrogate model is needed. This benefit is tied to the fact that the design points are quadrature nodes and those true model responses obtained during the surrogate model building process can be combined with quadrature weights to yield desired quantities as those quantities such as statistical moments and Sobol' indices are no more than integrals which are primary computational goals of the moment quadrature method in the first place. For completeness, the formulations for statistical moment evaluation and global sensitivity analysis based on the MQDP surrogate are provided in the Appendix. These procedures have been thoroughly investigated and validated in previous studies. Specifically, the computation of moments using moment quadrature techniques was systematically examined in [39], and the evaluation of Sobol' global sensitivity indices was detailed in [48], where extensive benchmark examples and comparisons with analytical solutions and Monte Carlo-based results were reported.

It is worth noting that the curse of dimensionality poses a fundamental challenge for high-dimensional surrogate modeling. For a full tensor product k-node quadrature rule in d dimensions, the number of design points scales as $N_{\text{full}} = k^d$, which becomes intractable even for moderate d. In contrast, the sparse constructions based on the Smolyak rule can significantly reduce the growth rate to polynomial order of $\propto d^L$ with a level parameter L. In the extreme case of $L = 1$, the number of design points for sparse MQDP is $2d + 1$, being a linear scaling. The detailed derivation of the scaling factor of the two can be found in, e.g. [39]. This reduction enables practical surrogate construction for high-dimensional problems when low-to-moderate interaction orders dominate the response behavior. The sparse MQDP strategy adopted in this work leverages this property, making surrogate modeling feasible for problems with tens to hundreds of random variables, as demonstrated in the numerical examples of section 4.

Post-trimming can be performed to drop polynomial terms whose coefficients are negligible in the built surrogate. In the present study, after constructing the polynomial surrogate, the absolute value of each coefficient is normalized with respect to the maximum coefficient magnitude in the surrogate. Terms whose normalized magnitude falls below a prescribed tolerance of 10^{-8} are regarded as numerically trivial and removed.

4. Benchmark and application examples

The accuracy and efficiency of the proposed method are demonstrated using standard benchmarking problems provided by the Surrogate Modeling Toolbox (SMT) [49]. The problem set offers a variety of problems used by many surrogate modeling studies.

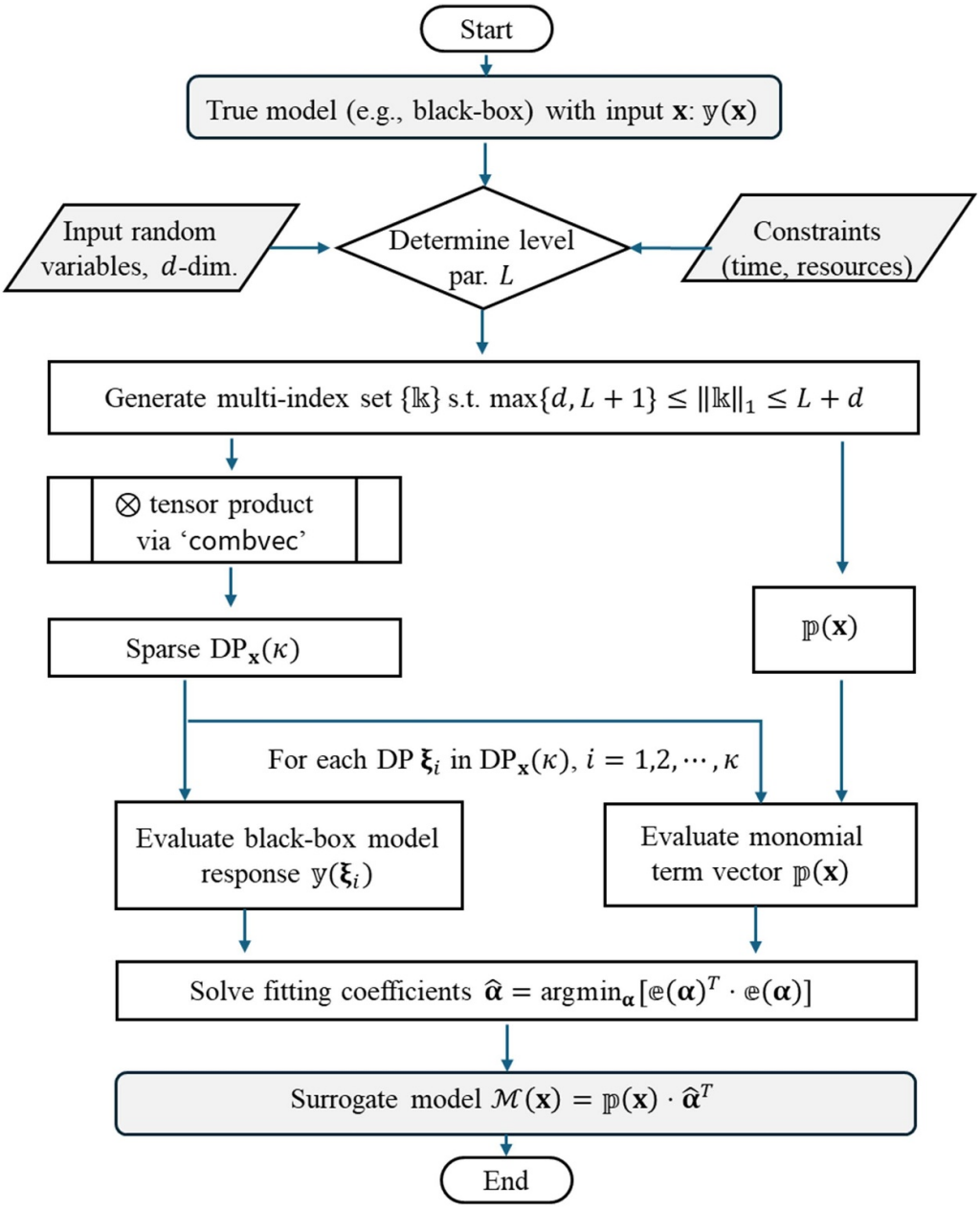
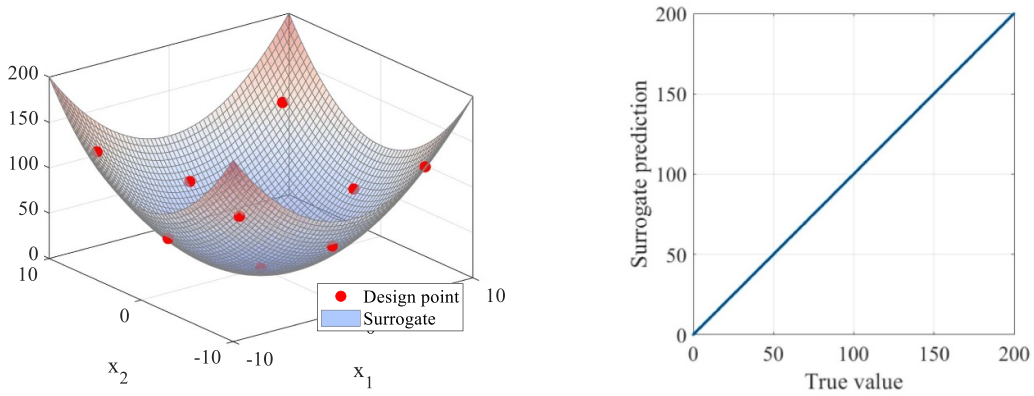


Figure 5. The standard procedure of sparse MQDP surrogate model construction.

4.1. Sphere function

The d-dimensional sphere function, i.e. the true model, defined as $\mathbb{y}(\mathbf{x}) = \sum_{i=1}^{d} x_i^2$ with $d = 2$ and $x_1, x_2 \in [-10, 10]$ is examined using standard MQDP with $k_1 = k_2 = 3$. A total number of 9 MQDPs are obtained via $(-7.746, 0, 7.746) \otimes (-7.746, 0, 7.746)$, where the two vectors are MQ nodes acquired following section 2. The monomials $p(\mathbf{x})$ are identified via $\left(1, x_1, x_1^2\right) \otimes \left(1, x_2, x_2^2\right)$ as $\left\{1, x_1, x_1^2, x_2, x_1x_2, x_1^2x_2, x_2^2, x_1x_2^2, x_1^2x_2^2\right\}$. The model fitting coefficients are obtained using equation (28) as $\boldsymbol{\alpha} = [0,0,1,0,0,0,1,0,0]$, and the resulting surrogate model is $\mathcal{M}(\mathbf{x}) = p(\mathbf{x})\,\alpha^T = x_1^2 + x_2^2$ which restores the true model. The performance of the built surrogate model is studied using a random set of MC samples. The surrogate model prediction results for the random validation points are compared with those obtained using the true model. Figure 6(a) shows the built surrogate model and the MQDPs, and figure 6(b) presents the prediction performance of the built model. In this case, the MQDP surrogate restores the true model and thus there is no deviation between the prediction results and the true values.



(a) (b)

Figure 6. (a) The built surrogate and MQDPs, and (b) model prediction vs. true value for 10^6 random points. Sphere function.

4.2. *Branin function*

The Branin function is given by

$$\mathrm{y}(\mathbf{x}) = \left(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos(x_1) + 10, \tag{49}$$

where $\mathbf{x} = (x_1, x_2)$ with $-5 \leqslant x_1 \leqslant 10$ and $0 \leqslant x_2 \leqslant 15$. The MQDP model was built with $k_1 = k_2 = 10$. Figure 7(a) presents the built surrogate model with MQDPs, and figure 7(b) shows the model prediction and the true values. The standard deviation of the model prediction residuals is 0.4362. A refined model with $k_1 = k_2 = 13$ can further reduce the standard deviation to 0.054. It is noted that the Branin function cannot be fully restored due to the existence of the trigonometric function, but the prediction error can asymptotically approach zero as the number of MQDPs increases.

4.3. *ℓ_p-norm function*

The ℓ_p -norm of a vector is defined as

$$\mathrm{y}(\mathbf{x}) = \left(\sum_{i=1}^{d} |x_i|^p\right)^{1/p}, \tag{50}$$

where $\mathbf{x} = [x_1, x_2]$, $p = 3$, and $x_1, x_2 \in [-1, 1]$. Figures 8(a) and (b) present the built MQDP surrogate model with $k_1 = k_2 = 7$ and the MQDPs, and the prediction results and true values for 10^6 random points, respectively. Deviations can be seen from figure 8(b) resulting from the fluctuation of the polynomial surface.

4.4. *20-dimensional Rosenbrock function*

The 20-variate Rosenbrock function is expressed as

$$\mathrm{y}(\mathbf{x}) = \sum_{i=1}^{20}\left[\left(x_{i+1} - x_i^2\right)^2 + (x_i - 1)^2\right], \tag{51}$$

where $x_i \in [-2, 2]$ for $i = 1, \ldots, 20$. Here the dimension is too large for the regular MQDP model. Even for $k_i = 2$ the number of MQDPs would be 2^{20}. Therefore, the sparse MQDP model with a level parameter $L = 4$ is used. In this case, the number of elements in the multi-index set $20 \leqslant |\mathbb{k}|_1 \leqslant 24$ is 10 626, which corresponds to the number of monomials in $p(\mathbf{x})$. The number of sparse DPs is 135 751, and the resulting number of fitting coefficients that are larger than 10^{-8} is only 78. In fact, using the level parameter $L = 4$ can fully restore the true function as the true function in equation (51) is indeed a polynomial with 78 terms. The model prediction results for 10^6 random points in the 20-variate space and the true values are shown in figure 9(a). For comparison, the MQDP surrogate model with $L = 3$ cannot fully restore the polynomial and deviations are shown in figure 9(b). In the latter case, the total number of monomials in $p(x)$ is 1771 and after trimming the number of effective terms is 59, and the total number of sparse MQDPs is 12 341.

4.5. *Tensor product functions*

The four product functions are examined.

$$\mathrm{y}(\mathbf{x}) = \begin{cases} \prod_{i=1}^{d} \cos(\pi x_i), -1 \leqslant x_i \leqslant 1 \\ \prod_{i=1}^{d} \exp(x_i), -1 \leqslant x_i \leqslant 1 \\ \prod_{i=1}^{d} \tanh(x_i), -1 \leqslant x_i \leqslant 1 \\ \prod_{i=1}^{d} \exp\left(-2x_i^2\right), -1 \leqslant x_i \leqslant 1 \end{cases}, \tag{52}$$

where $\mathbf{x} = x_1, x_2, \ldots, x_d$. To visualize the functions, $d = 2$ is tested. The results are shown in figure 10 where the comparisons with true values show the built MQDP surrogate models are accurate to capture the variations in those nonlinear functions. The numbers of MQDPs for the four functions of 'cos', 'exp', 'tanh', and 'Gaussian' are 49, 25, 25, and 49, respectively.

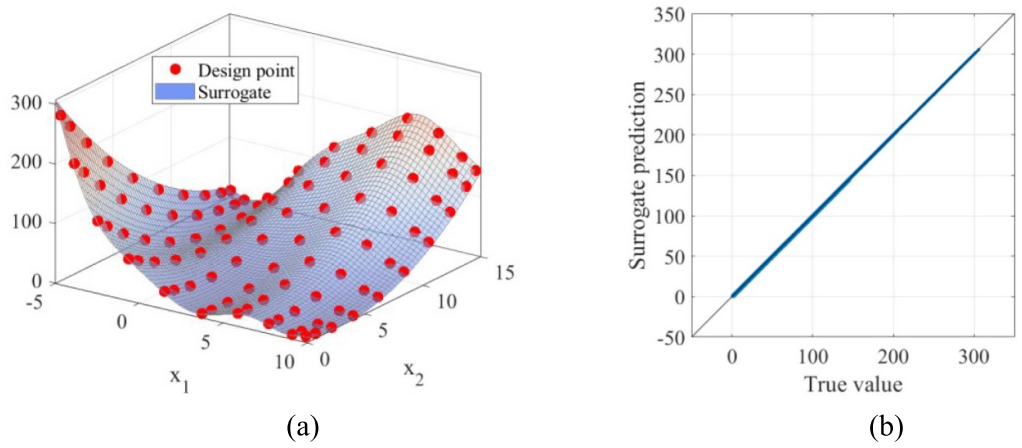


Figure 7. (a) The built surrogate and MQDPs, and (b) model prediction vs. true value for 10^6 random points. Branin function.

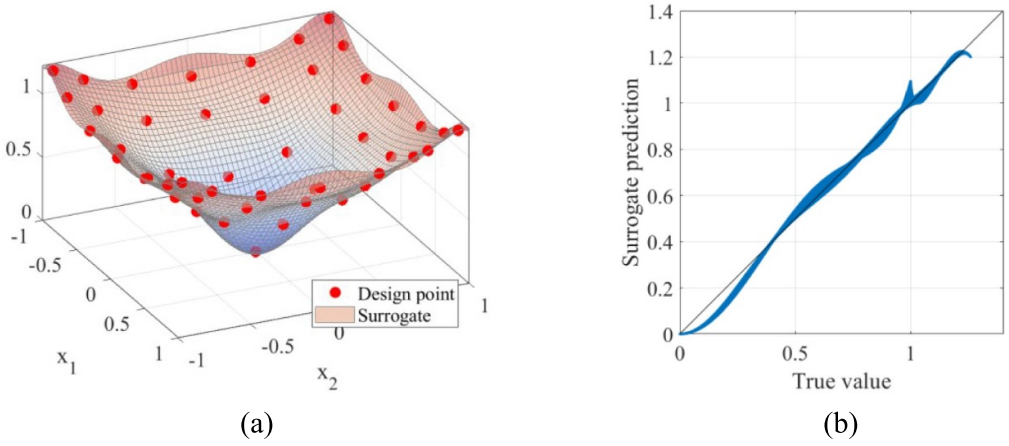


Figure 8. (a) The built surrogate and MQDPs, and (b) model prediction vs. true value for 10^6 random points. ℓ_p -norm function with $p = 3$.

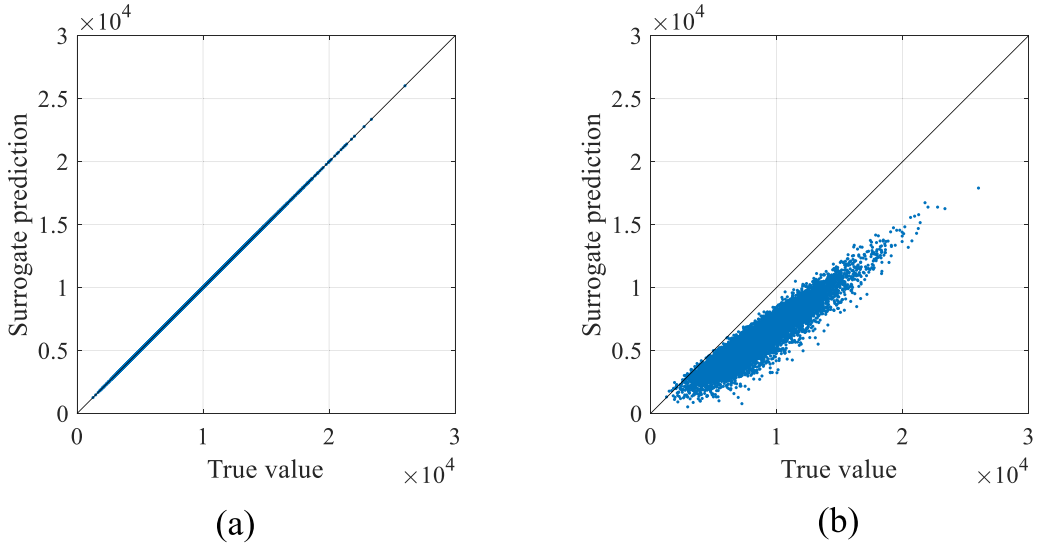


Figure 9. Results on model prediction vs. true value for 10^6 random points. 20-variate Rosenbrock function. Sparse MQDP surrogate models built with (a) L = 4, and (b) L = 3.

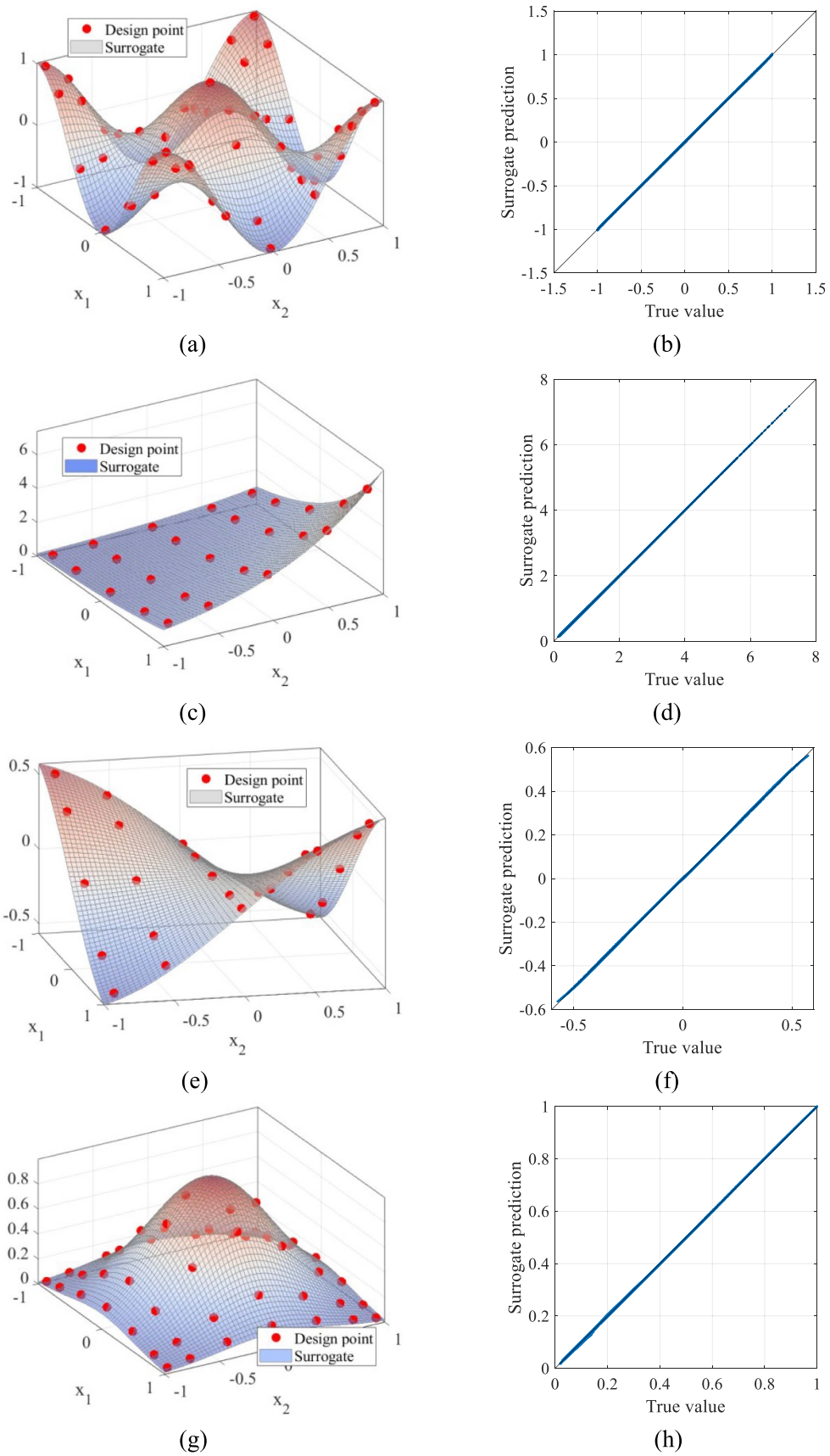


Figure 10. (a), (c), (e), (g) The built surrogate model and MQDPs, and (b), (d), (f), (h) model prediction vs. true value for random points for tensor product functions of 'cos', 'exp', 'tanh', and 'Gaussian', respectively.

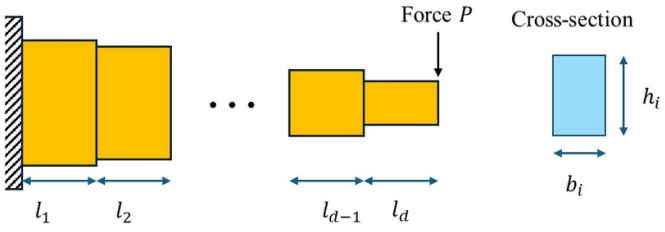


Figure 11. The schematic diagram of the 17-segment stepped cantilever beam.

4.6. A 51-variate stepped cantilever beam

The tip deflection of a 17-segment stepped cantilever beam shown in figure 11 is expressed as

$$\mathrm{y}(\mathbf{x}) = \frac{P}{3E}\sum_{i=1}^{d}\left\{\frac{12}{b_i h_i^3}\left[\left(\sum_{j=i}^{d} l_j\right)^3 - \left(\sum_{j=i+1}^{d} l_j\right)^3\right]\right\}, \quad (53)$$

where $d = 17$, $\mathbf{x} = [b_1, h_1, l_1, \ldots, b_d, h_d, l_d]$ is the random vector, and $P = 50$ kN and $E = 200$ GPa are the applied transverse force at the tip and Young's modulus, respectively. The width, height, and length of the ith rectangular cross-section segment are $b_i \in [0.04, 0.05]$, $h_i \in [0.3,\ 0.5]$, and $l_i \in [0.8,\ 1]$, respectively for $i = 1, \ldots, d$. The number of random variables is 51 in this case.

In this case, the sparse MQDP surrogate model is built with a level parameter of $L = 2$, which generates a total number of 5356 MQDPs and a total number of 1378 monomial terms. To verify the performance of the surrogate model, 10^4 random points are generated and tested. Figure 12(a) presents the results on model prediction vs. true value for the random points, and figure 12(b) shows the empirical PDF of the model prediction residuals. The standard deviation of the model prediction residuals is 0.0034, implying that the built surrogate model can offer accurate prediction results.

4.7. Mixed cantilever beam

The mixed cantilever beam has twelve distinct cross-sections with each associated with a dimensionless second moment of area $\tilde{I}$ about its neutral axis. The tip deflection can therefore be expressed as

$$\mathrm{y}(\mathbf{x}) = \frac{P}{3E}\frac{L^3}{S^2\tilde{I}}, \quad (54)$$

where $\mathbf{x} = [L, S]$ with $L \in [10, 20]$ and $S \in [1, 2]$, and $\tilde{I} \in \{0.0833,\ 0.139,\ 0.38,\ 0.0796,\ 0.133,\ 0.363,\ 0.0859,\ 0.136,\ 0.36,\ 0.0922,\ 0.138,\ 0.369\}$. Using $k_1 = k_2 = 4$, the 16 MQDPs and the built model are shown in figure 13(a). Figure 13(b) presents the validation results on the surrogate predictions vs. the true values of 10^6 random testing points, where a close agreement between the two is shown.

4.8. 8-variate robot arm problem

The robot arm function is a widely used problem in neural network studies. The function models a 4-segment planar manipulator whose shoulder arm is fixed at the origin of the plane. The segments have lengths L_1, L_2, L_2, and L_4. The first segment is rotated by an absolute angle θ_1 from the horizontal, and segment t is rotated by θ_t relative to segment $t-1$ for $t = 2, 3, 4$. The end position of the robot arm can then be expressed as

$$\mathrm{y}(\mathbf{x}) = \sqrt{\left[\sum_{i=1}^{4} L_i \cos\left(\sum_{j=1}^{i}\theta_j\right)\right]^2 + \left[\sum_{i=1}^{4} L_i \sin\left(\sum_{j=1}^{i}\theta_j\right)\right]^2}, \quad (55)$$

where $L_i \in [0, 1]$ for $i = 1, \ldots, 4$ and $\theta_j \in [0, 2\pi]$ for $j = 1, \ldots, 4$. The total number of random variables is 8. A regular MQDP surrogate model is built with $k_t = 4$ for $t = 1, \ldots, 8$. This setup generates 65 535 MQDPs and the resulting surrogate model prediction vs. true values for 10^3 random points are presented in figure 14(a), and the empirical PDF of the model prediction residuals is shown in figure 14(b).

4.9. 15-variate torsion vibration problem

The torsional vibration frequency of a disk-shaft rotor shown in figure 15(a) is determined by a 15-variate function,

$$\mathrm{y}(\mathbf{x}) = \frac{1}{2\pi}\left(\frac{-b-\sqrt{b^2-4ac}}{2a}\right)^{1/2}, \quad (56)$$

where $a = 1$, $b = -\left(\frac{K_1+K_2}{J_1} + \frac{K_2+K_3}{J_2}\right)$, $c = \frac{K_1K_2+K_2K_3+K_3K_1}{J_1J_2}$, $K_i = \frac{\pi G_i d_i}{32L_i}$ for $i = 1, 2, 3$, and $M_j = \frac{\pi}{4g}\rho_j t_j D_j$ and $J_j = \frac{1}{8}M_j D_j^2$ for $j = 1, 2$. The term $g = 9.8$ is constant. The nominal values of the disk and shaft parameters are given in table 1. The parameters are uniform random variables centered at their nominal values with a 5% maximum deviation amplitude around the nominal values, e.g. $d_1 \in [1.9,\ 2.1]$ etc. The assumed deviation range reflects the potential manufacturing uncertainty. The random variable $\mathbf{x} = [d_1, L_1, G_1, d_2, L_2, G_2, d_3, L_3, G_3, D_1, t_1, \rho_1, D_2, t_2, \rho_2]$.

The sparse MQDP surrogate model is built with the level parameter $L = 1$, resulting in a total of 31 sparse MQDPs. The built surrogate model is a 16-term linear model. Figures 15(a) and (b) show the model predictions vs. the true values of 10^3 random points, and the empirical PDF of the prediction residuals, respectively. In this case, the surrogate model performs well, and the overall prediction error is below 1% of the nominal frequency.

4.10. Water flow rate problem

The flowrate of water through a borehole that is drilled from the ground surface through two aquifers is described by the following equation,

$$\mathrm{y}(\mathbf{x}) = \frac{2\pi T_u(H_u - H_l)}{\ln(r/r_w)\left[1 + \frac{2LH_u}{\ln(r/r_w)r_w^2 K_w} + \frac{T_u}{T_l}\right]}, \quad (57)$$

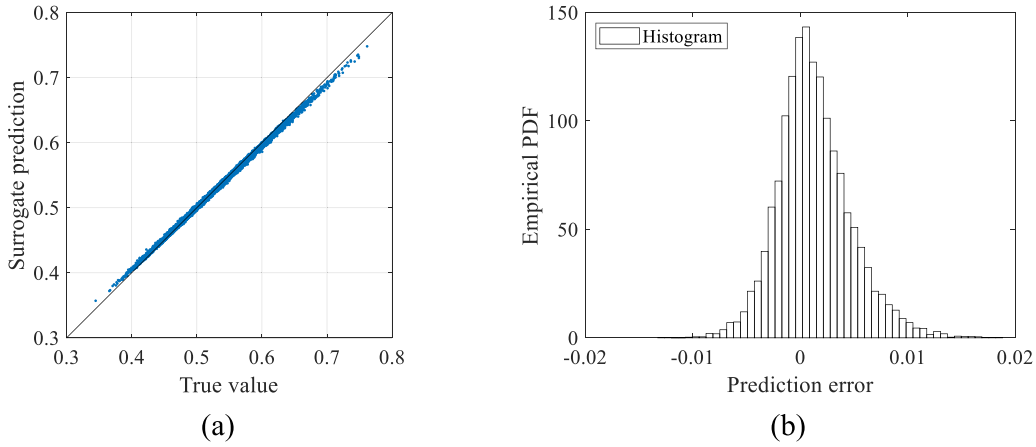


Figure 12. (a) Results on model prediction vs. true value for 10^4 random points, and (b) the histogram of the model prediction residuals. Stepped cantilever beam function.

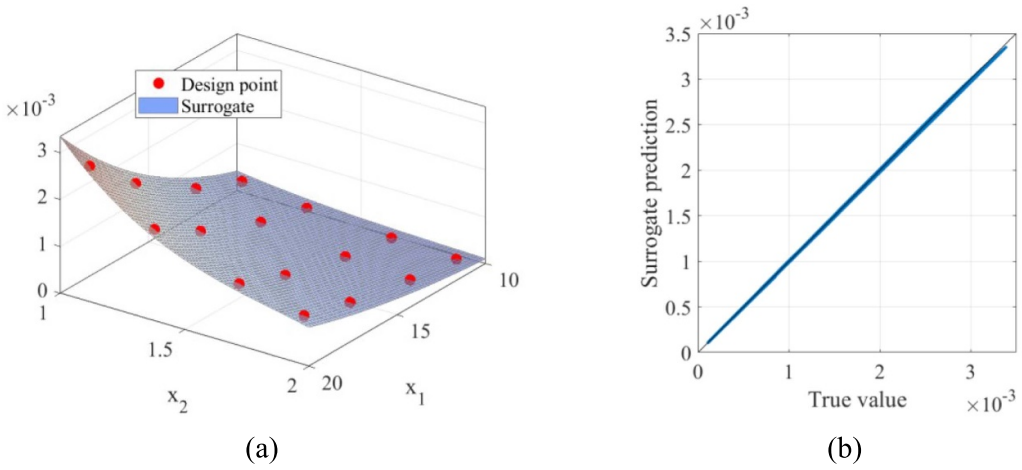


Figure 13. (a) The built surrogate and MQDPs, and (b) model prediction vs. true value for 10^6 random points. Mixed cantilever beam.

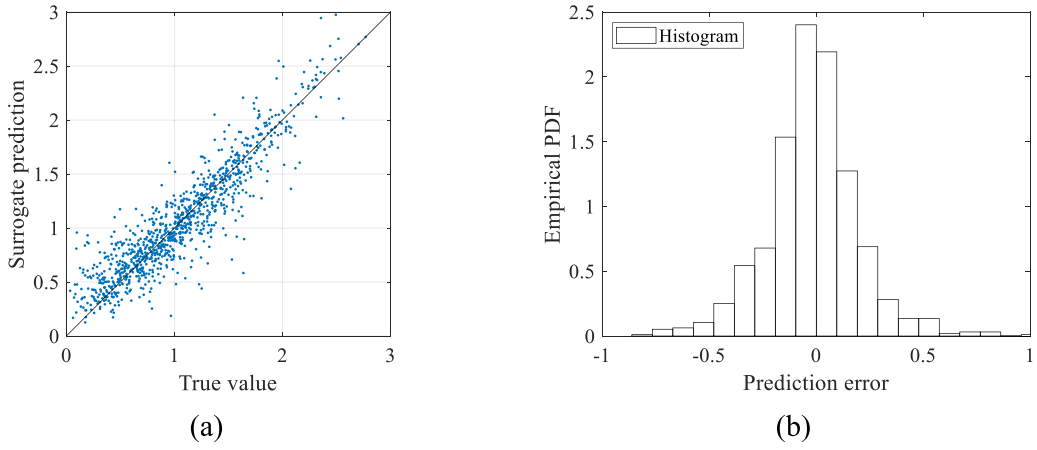


Figure 14. (a) Results on model prediction vs. true value for 10^3 random points, and (b) the histogram of the model prediction residuals. Robot arm function.

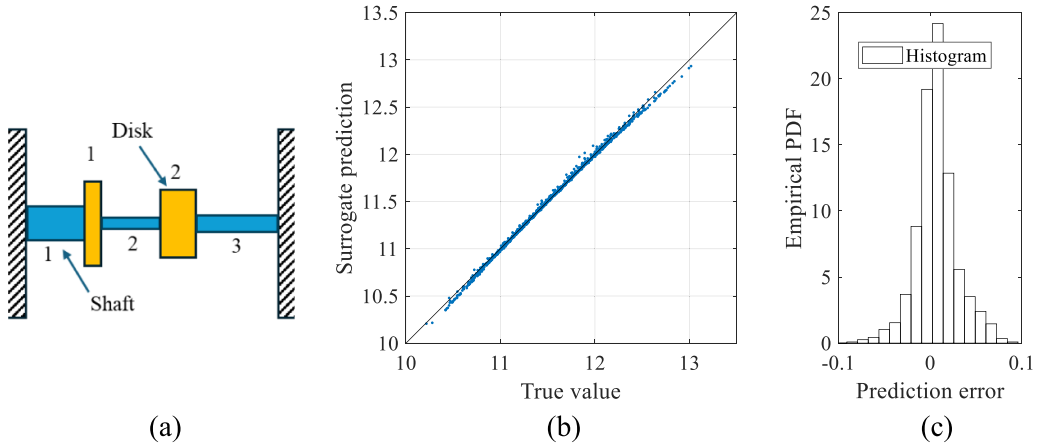


Figure 15. (a) Schematic diagram of the disk-shaft rotor, (b) results on model prediction vs. true value for 10^3 random points, and (c) the histogram of the model prediction residuals. Torsional vibration function.

Table 1. The nominal values of the shaft and disk parameters.

Parameter	Shaft 1	Shaft 2	Shaft 3	Disk 1	Disk 2
Diameter (in.)	$d_1 = 2$	$d_2 = 1.825$	$d_3 = 2.25$	$D_1 = 12$	$D_2 = 14$
Length (in.)	$L_1 = 10$	$L_2 = 12$	$L_3 = 8$	$t_1 = 3$	$t_2 = 4$
Modulus, G (lb/in.^{2}) Density, ρ, (lb/in.^{3})	$G_1 = 11.7 \cdot 10^6$	$G_2 = 6.2 \cdot 10^6$	$G_3 = 3.9 \cdot 10^6$	$\rho_1 = 0.28$	$\rho_2 = 0.1$

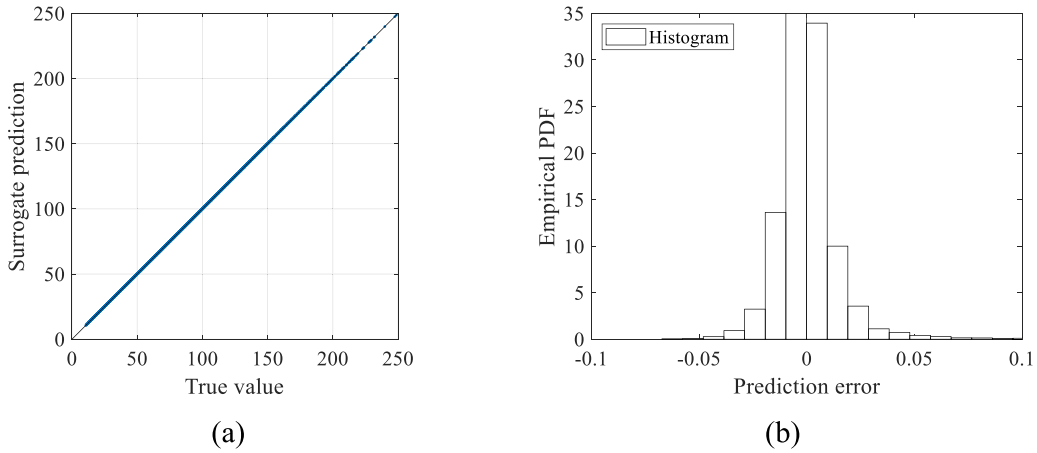


Figure 16. (a) Results on model prediction vs. true value for 10^4 random points, and (c) the histogram of the model prediction residuals. Water flowrate function.

where the eight input random variables and their ranges are the radius of borehole, $r_w \in [0.05, 0.15]$ m, the radius of influence, $r \in [100, 50000]$ m, the transmissivity of upper aquifer, $T_u \in [63070, 115600]$ m^2 year^{-1}, the potentiometric head of upper aquifer, $H_u \in [990, 1110]$ m, the transmissivity of lower aquifer, $T_l \in [63.1, 116]$ m^2 year^{-1}, the potentiometric head of lower aquifer, $H_l \in [700, 820]$ m, the length of borehole, $L \in [1120, 1680]$ m, and the hydraulic conductivity of borehole, $K_w \in [9855, 12045]$ m year^{-1}. The random variables form the 8-variate random vector $\mathbf{x} = [r_w, r, T_u, H_u, T_l, H_l, L, K_w]$. Using the regular MQDP surrogate model construction with $k_i = 4$ for $i = 1, \ldots, 8$, a total number of 65 536 MQDPs are generated. The built model is tested with 10^4 random points, and model prediction results are compared with the true values. Results are shown in figure 16(a), and the empirical PDF of model prediction residuals is shown in figure 16(b). The built model can yield very accurate predictions.

4.11. Welded beam problem

The welded beam problem is widely used in genetic algorithms literature. The shear stress of the welded beam is expressed as

$$y(\mathbf{x}) = \sqrt{\tau'^2 + \tau''^2 + l\tau'\tau'' / \sqrt{0.25\left[l^2 + (h+t)^2\right]}}, \tag{58}$$

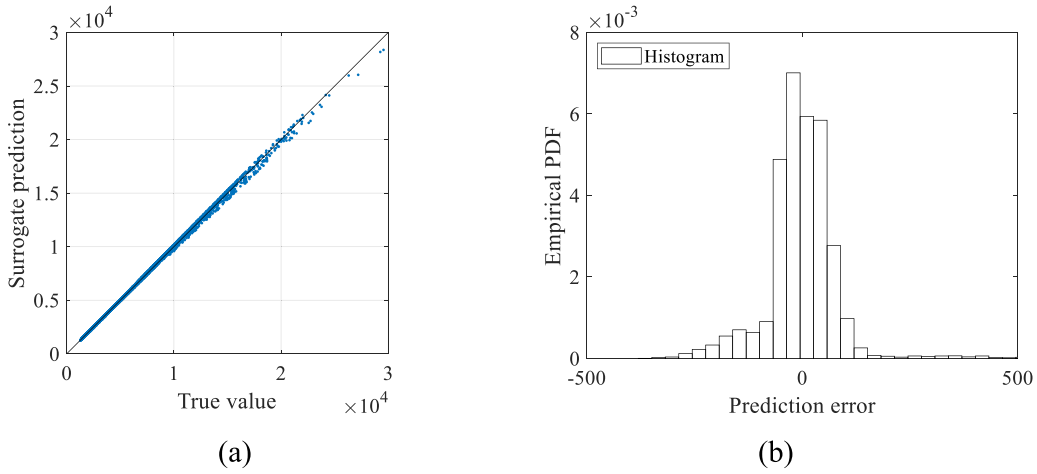


Figure 17. (a) Results on model prediction vs. true value for 10^4 random points, and (c) the histogram of the model prediction residuals. Welded beam stress function.

where $\tau' = \frac{6000}{\sqrt{2}hl}$ and $\tau'' = \frac{6000(14+l/2)\sqrt{0.25\left[l^2+(h+t)^2\right]}}{\sqrt{2}hl\left[l^2/12+0.25(h+t)^2\right]}$. The three design variables $\mathbf{x} = [h, l, t]$ are $h \in [0.125,\ 1]$, $l \in [5,\ 10]$, and $t \in [5,\ 10]$. A 6-node MQ rule is used to generate a total number of 216 MQDPs. The built MQDP surrogate model has 216 effective terms, and the model prediction results on 10^4 random points are compared with true values and shown in figure 17. Overall, the surrogate model demonstrates satisfactory performance in the random test.

4.12. Aircraft wing weight function

The following analytical expression is used as a conceptual level estimate for the aircraft wing weight,

$$\begin{aligned} y(x) = {} & 0.036 S_w^{0.785} W_{fw}^{0.0035} \left(\frac{A}{\cos^2\Lambda}\right)^{0.6} q^{0.006} \lambda^{0.04} \\ & \times \left(\frac{100 R_{tc}}{\cos\Lambda}\right)^{-0.3} (N_z W_{dg})^{0.49} + S_w W_p, \end{aligned} \tag{59}$$

where the 10 random variables are detailed in table 2.

Here the sparse MQDP surrogate model is built with a level parameter of $L = 3$. A total number of 1771 design points are generated, and the number of effective monomial terms in the surrogate model is 117. The built model is used to predict 10^4 random points in the design space, and the results are compared with true values. The comparisons and the empirical PDF of the model prediction residuals are presented in figures 18(a) and (b), respectively. The mean weight is 626.3 lb while the standard deviation of the prediction residuals is 6.65 lb which is about 2 orders smaller, implying the model can in general yield accurate predictions.

4.13. Ishigami function

The Ishigami function is a standard testing function in sensitivity analysis. Despite its simple form, the function is considered highly nonlinear and the surrogate modeling on this function is challenging. The Ishigami function writes

$$y(\mathbf{x}) = \sin(x_1) + 7\sin^2(x_2) + 0.1 x_3^4 \sin(x_1), \tag{60}$$

where $\mathbf{x} = [x_1, x_2, x_3]$ and $x_1, x_2, x_3 \in [-\pi, \pi]$. The regular MQDP surrogate model is built with $k_1 = k_2 = k_3 = 10$, rendering a total number of 1000 MQDPs. The built model is tested with 10^4 random points, and the comparison between the model prediction and true values is shown in figure 19, where a close agreement between the two can be observed.

4.14. 100-dimensional function

The so-called 100D function from UQLab[1] is expressed as

$$\begin{aligned} y(\mathbf{x}) = {} & 3 - \frac{5}{d}\sum_{i=1}^{d} i x_i + \frac{1}{d}\sum_{i=1}^{d} i x_i^3 + \frac{1}{3d}\sum_{i=1}^{d} i \ln\left(x_i^2 + x_i^4\right) \\ & + x_1 x_2^2 + x_2 x_4 - x_3 x_5 + x_{51} + x_{50} x_{54}^2, \end{aligned} \tag{61}$$

where $\mathbf{x} = [x_1, \ldots, x_{100}]$, $x_i \in [1,2]$, $i \neq 20$, $x_{20} \in [1,3]$, and $d = 100$. The sparse MQDP model is built using a level parameter of $L = 2$, resulting in 20 301 MQDPs and 5151 effective monomial terms in the sparse MQDP surrogate model. A total number of 10^4 random points in the 100-variate space are generated for model performance verification, and the model predictions of the random input points are compared with the true values obtained using equation (61). The results are presented in figure 20(a), and the empirical PDF of the model prediction residuals is shown in figure 20(b). The results demonstrate the effectiveness and accuracy of the sparse MQDP surrogate model. This example is intended to demonstrate the feasibility and robustness of the sparse MQDP-based surrogate framework in a genuinely high-dimensional setting. Although the absolute number of design points remains non-trivial, it represents a substantial reduction compared to full tensor-product constructions and enables stable surrogate reconstruction without adaptive refinement or problem-specific tuning.

[1] www.uqlab.com/sensitivity-high-dimension.

Table 2. The random variables and their ranges.

x_i	Parameter	Symbol	Range
1	Wing area (ft^2)	S_w	[150, 200]
2	Weight of fuel in wing (lb)	W_{fw}	[220, 300]
3	Aspect ratio	A	[6, 10]
4	Quarter-chord sweep (deg.)	Λ	[−10, 10]
5	Dynamic pressure at cruise (lb/ft^2)	q	[16, 45]
6	Taper ratio	λ	[0.5, 1]
7	Aerofoil thickness to chord ratio	R_{tc}	[0.08, 0.18]
8	Ultimate load factor	N_z	[2.5, 6]
9	Flight design gross weight (lb)	W_{dg}	[1700, 2500]
10	Paint weight (lb/ft^2)	W_p	[0.025, 0.08]

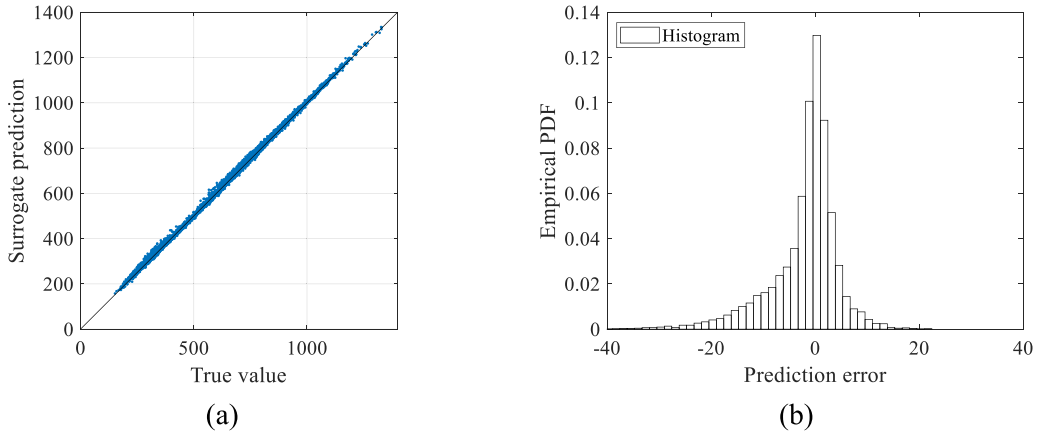


Figure 18. (a) Results on model prediction vs. true value for 10^4 random points, and (c) the histogram of the model prediction residuals. Wing weight function.

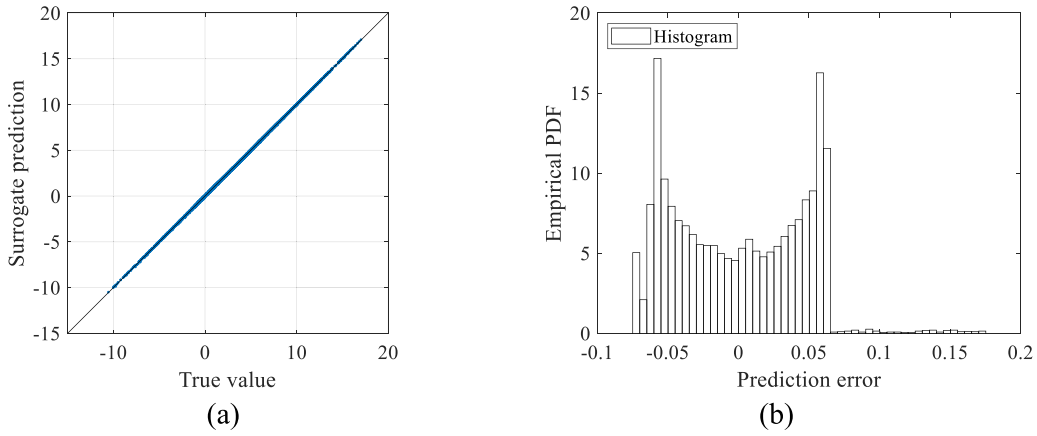


Figure 19. (a) Results on model prediction vs. true value for 10^4 random points, and (c) the histogram of the model prediction residuals. Ishigami function.

The results therefore highlight the scalability of the proposed deterministic methodology rather than optimal performance under extreme computational cost constraints.

It is important to clarify that the proposed MQDP-based surrogate modeling framework is not a PCE-based method. While both approaches employ polynomial representations, classical PCE constructs an orthogonal polynomial expansion of the model response and determines expansion coefficients via projection or regression techniques. In contrast, the MQDP framework treats moment quadrature nodes as deterministic design points for direct polynomial surrogate construction on monomial bases. The quadrature structure ensures optimal polynomial exactness of statistical moment evaluation, rather than serving as a projection tool for orthogonal expansion coefficients. Consequently, MQDP-based surrogate modeling constitutes a deterministic quadrature-guided polynomial interpolation framework rather than a PCE methodology.

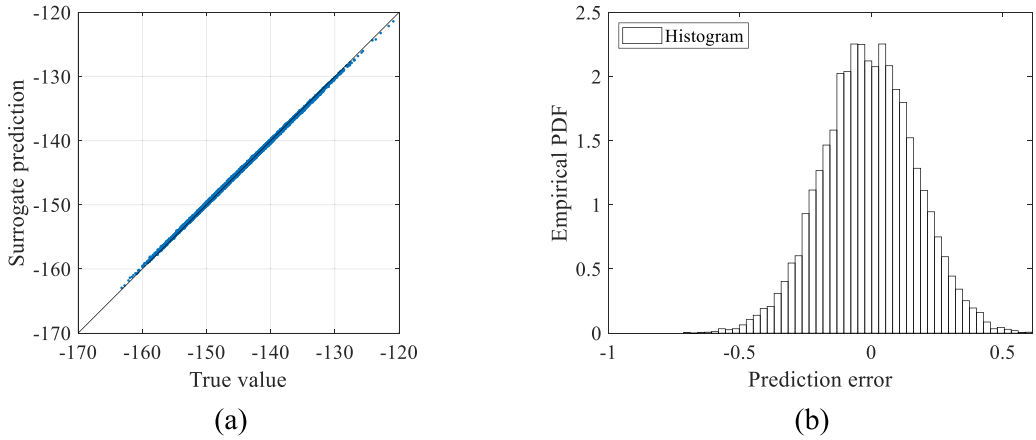


Figure 20. (a) Results on model prediction vs. true value for 10^4 random points, and (c) the histogram of the model prediction residuals. 100D function from UQLab.

Kernel-based surrogate models, such as Kriging or GP regression, rely on fundamentally different approximation mechanisms and modeling assumptions, including kernel selection and hyperparameter optimization. It should be noted that the objective of the present work is to establish a deterministic surrogate framework with guaranteed polynomial exactness properties. A comprehensive performance comparison with kernel-based surrogate models such as GP regression is beyond the scope of this study and will be considered in future investigations.

While polynomial-based approximations may suffer from ill-conditioning when very high polynomial orders or excessively large basis sets are used, numerical stability in the present framework is maintained by adopting moderate quadrature levels, sparse-grid MQDP constructions, and optional pruning of negligible polynomial terms. Regularization techniques are not explicitly employed in this study, as no numerical instability was observed for the considered examples. The incorporation of conditioning control and regularization strategies for more challenging scenarios is a subject of future work.

5. Conclusions

The study developed a new methodology for constructing surrogate models via MQDPs. The mathematical features of the MQDPs assure a deterministic and definitive procedure to be established. Unlike traditional random or semi-random sampling approaches, the proposed framework leverages the exactness properties of Gauss quadrature to guarantee a theoretically justified basis selection and numerical stability for model construction. The methodology is general, fully algorithmic, and applicable to a broad range of PDFs, including nonuniform and empirical distributions. Extensive tests on analytical benchmarks, high-dimensional synthetic problems, and engineering applications, including a 51-variate stepped cantilever beam and a 100-dimensional function, demonstrated the accuracy, computational efficiency, and scalability of the proposed framework. For example, the sparse MQDP surrogate for the 20-dimensional Rosenbrock function successfully recovered the exact polynomial structure using 135 751 design points while retaining only 78 effective monomial terms. In the 51-variate stepped cantilever beam problem, accurate predictions were achieved with a sparse construction using 5356 design points. The 100-dimensional benchmark example further demonstrated stable surrogate reconstruction with 20 301 MQDPs and 5151 effective monomial terms, highlighting the scalability of the deterministic sparse framework. Based on current results, the following conclusions can be drawn.

(1) A unified MQDP generation scheme is developed to deterministically produce design points for surrogate modeling under prescribed PDFs. The proposed approach avoids the arbitrariness associated with random sampling strategies such as LHS and remains applicable to nonuniform and empirical input distributions.

(2) The proposed framework directly exploits the algebraic exactness of Gauss quadrature. When polynomial basis functions up to the quadrature order are employed, surrogate models constructed on MQDPs recover statistical moments exactly, including mean, variance, skewness, and kurtosis. This property establishes MQDP-based surrogates as an optimal polynomial approximation strategy for a given number of design points.

(3) The univariate MQDP concept is systematically extended to multi-dimensional problems through tensor-product formulations, with linear correlation among input variables incorporated via covariance-based transformations. To mitigate the curse of dimensionality, a sparse MQDP construction based on the Smolyak rule is introduced, enabling efficient surrogate modeling for high-dimensional problems. Numerical examples demonstrated stable performance in problems with up to 100 random variables, while maintaining controlled basis growth and accurate prediction capability.

(4) The methodology provides an automated pipeline for surrogate construction, including monomial basis generation, coefficient regression, and optional pruning of negligible terms. Owing to the quadrature nature of MQDPs, the same set of model evaluations can also support uncertainty quantification and global sensitivity analysis without additional computations, thereby improving overall computational efficiency.

In the present work, dependence among input variables is treated using covariance-based transformations, which are suitable for modeling linear correlation structures. General nonlinear correlations, such as those described by non-Gaussian copulas or higher-order dependence, cannot be directly accommodated within the current framework. Extending the proposed approach to incorporate nonlinear dependence modeling is an important direction for future research. Moreover, structural response field problems, which involve spatially distributed outputs and additional discretization or model reduction steps, are beyond the scope of the present study and will be investigated in future work. While the moment quadrature formulation does not require an explicit analytical PDF and can, in principle, be constructed from empirical moments, the present study does not include numerical examples based on empirical input distributions. In such cases, the accuracy of high-order empirical moments and the positive definiteness of the associated Hankel matrix may affect the robustness of the quadrature construction, and their systematic treatment is left for future work.

Acknowledgment

The work in this study was supported by Science Challenge Project, No. TZ2025017, and National Natural Science Foundation of China, No. U2230204. The support is gratefully acknowledged.

Appendix

A.1. Mean, variance, and high-order moments for uncertainty quantification

The kth-order raw statistical moment of the true model can be obtained using the full-tensor MQDPs and their associated weights as,

$$E\left[p^k(\mathbf{x})\right] \approx \sum_{i=1}^{\kappa} \mathbf{w}_i w^k(\mathbf{x}_i), \tag{62}$$

where $\mathbf{x}^T = [x_1, x_2, \ldots, x_d]$, and $\mathbf{x}_i$, $i = 1, \ldots, \kappa$ is the ith element in the set formed by the tensor product of the univariate MQDP for each random variable in the d-dimensional variable space, and $\mathbf{w}_i$ is the composite weight of the ith element, computed as the product of the nodal weights. The detailed mathematical background and derivations can be found in, e.g. [37, 38].

For sparse version, the kth-order raw statistical moment is formulated as [39]

$$E\left[y^k(x)\right] \approx \sum_{l=\max\{d,L+1\}}^{L+d} \sum_{\substack{\|K\|_1=l \\ K\in N^d, K\geqslant 1}} \left[(-1)^{L+d-l}\binom{d-1}{L+d-l} \sum_{i_1=1}^{k_1}\cdots\sum_{i_d=1}^{k_d} w^{(1)}_{k_1,i_1}\cdots w^{(d)}_{k_d,i_d}\, \mathcal{Y}\left(\xi^{(1)}_{k_1,i_1},\cdots\right)\xi^{(d)}_{k_d,i_d}\right], \tag{63}$$

where $\xi^{(i)}_{k_i,j}$ and $w^{(i)}_{k_i,j}$ are the jth node (i.e. MQDP) and weight of the k_i-node univariate MQ rule in the ith dimension, respectively, and the other symbols are defined as before.

A.2. Sobol' sensitivity indices for sensitivity analysis

The first-order Sobol' sensitivity indices are a decomposition of the variance of the model response for input variables. Sobol' showed that a d-dimensional function y with uniform input of $\mathbf{x} = [x_1, x_2, \ldots, x_d] \in U^d \equiv [0,1]^d$, can uniquely be decomposed into summands of increasing dimensionality such that,

$$\mathcal{Y}(\mathbf{x}) = y_0 + \sum_{i=1}^{d} y_i(x_i) + \sum_{1\leqslant i<j\leqslant d} y_{ij}(x_i, x_j) + \sum_{1\leqslant i<j<k\leqslant d} y_{ijk}(x_i, x_j, x_k) + y_{1,\ldots,d}(x_1, .., x_d), \tag{64}$$

given that the integral of each of the summands $y_{i_1,i_2,\ldots,i_s}(x_{i_1}, x_{i_2}, \ldots, x_{i_s})$ over any of its independent variables is zero

$$\int_U y_{i_1,i_2,\ldots,i_s}(x_{i_1}, x_{i_2}, \ldots, x_{i_s})\, \mathrm{d}x_{i_k} = 0, \forall 1 \leqslant k \leqslant s,\ k \in N. \tag{65}$$

The constant term y_0 is the mean of the model response,

$$y_0 = \int_{U^d} \mathcal{Y}(\mathbf{x})\, \mathrm{d}\mathbf{x}. \tag{66}$$

With above setting, the total variance by definition is

$$D = \int_{U^d} y^2(\mathbf{x})\, \mathrm{d}\mathbf{x} - y_0^2, \tag{67}$$

which can alternatively be expressed as

$$D = \sum_{i=1}^{d} D_i + \sum_{1\leqslant i\leqslant j\leqslant d} D_{ij} + \cdots + D_{1,2,\ldots,d}, \tag{68}$$

where the summand

$$D_{i_1,i_2,\ldots,i_s} = \int_{U^s} y^2_{i_1,i_2,\ldots,i_s}(x_{i_1}, \ldots, x_{i_s})\, \mathrm{d}x_{i_1}\ldots\mathrm{d}x_{i_s}. \tag{69}$$

The ratio

$$S_{i_1,i_2,\ldots,i_s} = \frac{D_{i_1,i_2,\ldots,i_s}}{D}, \tag{70}$$

is the so-called s-order Sobol' sensitivity index. With equation (69), the first-order term for the ith variable in the d-dimensional variable space is

$$D_i = \int_0^1 \left[\int_{[0,1]^{d-1}} \mathcal{Y}(\mathbf{x})\, \mathrm{d}\mathbf{x}_{\sim i} - y_0\right]^2 \mathrm{d}x_i, \tag{71}$$

where $\mathbf{x}_{\sim i} \in U^{d-1} \equiv [0,1]^{d-1}$ is the $(d-1)$ -dimensional random variable excluding the ith random variable. The second-order term involving the ith and jth variables, $i<j$, is

$$D_{ij} = \int_{[0,1]^2} \left[\int_{[0,1]^{d-2}} y(\mathbf{x})\, d\mathbf{x}_{\sim ij} - \int_{[0,1]^{d-1}} y(\mathbf{x})\, d\mathbf{x}_{\sim i} - \int_{[0,1]^{d-1}} y(\mathbf{x})\, d\mathbf{x}_{\sim j} + y_0 \right]^2 dx_i dx_j. \tag{72}$$

Similarly, the third-order term, for $i<j<k$, writes

$$D_{ijk} = \int_{[0,1]^3} \left[\int_{[0,1]^{d-3}} y(\mathbf{x})\, d\mathbf{x}_{\sim ijk} - \sum_{1\leqslant i<j\leqslant d} \int_{[0,1]^{d-2}} y(\mathbf{x})\, d\mathbf{x}_{\sim ij} + \sum_{1\leqslant i\leqslant d} \int_{[0,1]^{d-1}} y(\mathbf{x})\, d\mathbf{x}_{\sim i} - y_0 \right]^2 dx_i dx_j dx_k. \tag{73}$$

These integrals can be evaluated using the MQDPs with weights as

$$D_i \approx \sum_{\alpha_i=1}^{k_i} w_{\alpha_i} \left[\sum_{\alpha_1=1}^{k_1} \cdots \sum_{\alpha_{i-1}=1}^{k_{i-1}} \sum_{\alpha_{i+1}=1}^{k_{i+1}} \cdots \sum_{\alpha_d=1}^{k_d} w_{\alpha_1} \cdots w_{\alpha_{i-1}} w_{\alpha_{i+1}} \cdots w_{\alpha_d} y(\xi_{\alpha_1}, \ldots, \xi_{\alpha_d}) - y_0 \right]^2, \tag{74}$$

$$D_{ij} \approx \sum_{\alpha_i=1}^{k_i} \sum_{\alpha_j=1}^{k_j} w_{\alpha_i} w_{\alpha_j} \left[\begin{array}{l} \sum_{\alpha_1=1}^{k_1} \cdots \sum_{\alpha_{i-1}=1}^{k_{i-1}} \sum_{\alpha_{i+1}=1}^{k_{i+1}} \cdots \sum_{\alpha_{j-1}=1}^{k_{j-1}} \sum_{\alpha_{j+1}=1}^{k_{j+1}} \cdots \sum_{\alpha_d=1}^{k_d} w_{\alpha_1} \cdots w_{\alpha_d} y(\xi_{\alpha_1}, \ldots, \xi_{\alpha_d}) - \\ \sum_{\alpha_1=1}^{k_1} \cdots \sum_{\alpha_{i-1}=1}^{k_{i-1}} \sum_{\alpha_{i+1}=1}^{k_{i+1}} \cdots \sum_{\alpha_d=1}^{k_d} w_{\alpha_1} \cdots w_{\alpha_{i-1}} w_{\alpha_{i+1}} \cdots w_{\alpha_d} y(\xi_{\alpha_1}, \ldots, \xi_{\alpha_d}) - \\ \sum_{\alpha_1=1}^{k_1} \cdots \sum_{\alpha_{j-1}=1}^{k_{j-1}} \sum_{\alpha_{j+1}=1}^{k_{j+1}} \cdots \sum_{\alpha_d=1}^{k_d} w_{\alpha_1} \cdots w_{\alpha_{j-1}} w_{\alpha_{j+1}} \cdots w_{\alpha_d} y(\xi_{\alpha_1}, \ldots, \xi_{\alpha_d}) + y_0 \end{array} \right], \tag{75}$$

etc., where are the α_ith node and weight in the k_i-node MQ rule for the ith random variable. The detailed mathematical derivation of higher-order Sobol' sensitivity indices can be found in [48].

A.3. The minimum or maximum of the model

The minimum or maximum value of the built surrogate model can be found using a gradient-based iterative formula. As the built MQDP surrogate model is the summation of monomials, there exists closed-form arbitrary-order derivatives; therefore, the gradient of the built surrogate can readily be obtained using equation (26) as

$$\nabla M(\mathbf{x}) = \nabla p(\mathbf{x}) \cdot \boldsymbol{\alpha}^T. \tag{76}$$

With an initial guess $\mathbf{x}_0$ for seeking the minimum or maximum of $\mathcal{M}(\mathbf{x})$, the iteration sequence $\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2, \ldots$ follows

$$\mathbf{x}_{n+1} = \mathbf{x}_n - \lambda_n \nabla M(\mathbf{x}_n) = \mathbf{x}_n - \lambda_n \nabla p(\mathbf{x}_n) \cdot \boldsymbol{\alpha}^T, \tag{77}$$

where $\lambda_n \in \mathbb{R}^+$ is a small step size constant at nth iterative step, yielding a monotonic sequence $\mathcal{M}(\mathbf{x}_0) \geqslant \mathcal{M}(\mathbf{x}_1) \geqslant \mathcal{M}(\mathbf{x}_2) \geqslant \cdots$. When the model function $\mathcal{M}(x)$ is convex, the sequence converges to the global minimum value. Certain criteria can be set to stop the iteration by tracking the difference between two consecutive model runs. Moreover, the step size parameter is allowed to vary at each iteration to enhance the iteration efficiency. A good choice for λ given in [50] is

$$\lambda_n = \frac{\left| (x_n - x_{(n-1)})^T [\nabla \mathcal{M}(x_n) - \nabla \mathcal{M}(x_{n-1})] \right|}{\|\nabla \mathcal{M}(x_n) - \nabla \mathcal{M}(x_{n-1})\|^2} = \frac{\left| (x_n - x_{n-1})^T [\nabla \mathrm{p}(x_n) - \nabla \mathrm{p}(x_{n-1})] \cdot \alpha^T \right|}{\|[\nabla \mathrm{p}(x_n) - \nabla \mathrm{p}(x_{n-1})] \cdot \alpha^T\|^2}. \tag{78}$$

References

[1] Cui L-X, Du Y-M and Sun C-P 2025 On quantum reliability characterizing systematic errors in quantum sensing *J. Reliab. Sci. Eng.* **1** 015004

[2] Liu Z, Lesselier D, Sudret B and Wiart J 2020 Surrogate modeling based on resampled polynomial chaos expansions *Reliab. Eng. Syst. Saf.* **202** 107008

[3] Weng Y-Y, Liu T, Zhang X-Y and Zhao Y-G 2025 Probability density estimation of polynomial chaos and its application in structural reliability analysis *Reliab. Eng. Syst. Saf.* **253** 110537

[4] Shang X, Wang L, Fang H, Lu L and Zhang Z 2024 Active learning of ensemble polynomial chaos expansion method

for global sensitivity analysis *Reliab. Eng. Syst. Saf.* **249** 110226
[5] Kontolati K, Loukrezis D, Dos Santos K R, Giovanis D G and Shields M D 2022 Manifold learning-based polynomial chaos expansions for high-dimensional surrogate models *Int. J. Uncertainty Quantif.* **12** 39–64
[6] Liu J and Jiang C 2023 Surrogate modeling for high dimensional uncertainty propagation via deep kernel polynomial chaos expansion *Appl. Math. Modell.* **122** 167–86
[7] Sharma H, Novák L and Shields M 2024 Physics-constrained polynomial chaos expansion for scientific machine learning and uncertainty quantification *Comput. Methods Appl. Mech. Eng.* **431** 117314
[8] Mueller J N, Sargsyan K, Daniels C J and Najm H N 2025 Polynomial chaos surrogate construction for random fields with parametric uncertainty *SIAM/ASA J. Uncertain. Quantif.* **13** 1–29
[9] Zong C, Shi M, Li Q, Xue T, Song X, Li X and Chen D 2023 Sealing design optimization of nuclear pressure relief valves based on the polynomial chaos expansion surrogate model *Nucl. Eng. Technol.* **55** 1382–99
[10] Zhu X, Broccardo M and Sudret B 2023 Seismic fragility analysis using stochastic polynomial chaos expansions *Probab. Eng. Mech.* **72** 103413
[11] Zuhal L R, Zakaria K, Palar P S, Shimoyama K and Liem R P 2021 Polynomial-chaos–kriging with gradient information for surrogate modeling in aerodynamic design *AIAA J.* **59** 2950–67
[12] Marinescu M 2024 Explaining and connecting Kriging with Gaussian process regression (arXiv:2408.02331)
[13] Xu Y, Renteria A and Wang P 2022 Adaptive surrogate models with partially observed information *Reliab. Eng. Syst. Saf.* **225** 108566
[14] Marrel A and Iooss B 2024 Probabilistic surrogate modeling by Gaussian process: a new estimation algorithm for more robust prediction *Reliab. Eng. Syst. Saf.* **247** 110120
[15] Abaei M M, Leira B J, Sævik S and BahooToroody A 2024 Integrating physics-based simulations with gaussian processes for enhanced safety assessment of offshore installations *Reliab. Eng. Syst. Saf.* **249** 110235
[16] Wu J and Wan L 2024 Reliability sensitivity analysis for RBSMC: a high-efficiency multiple response Gaussian process model *Reliab. Eng. Syst. Saf.* **243** 109812
[17] Tran V N and Kim J 2022 Robust and efficient uncertainty quantification for extreme events that deviate significantly from the training dataset using polynomial chaos-kriging *J. Hydrol.* **609** 127716
[18] Lüthen N, Marelli S and Sudret B 2023 A spectral surrogate model for stochastic simulators computed from trajectory samples *Comput. Methods Appl. Mech. Eng.* **406** 115875
[19] Teng D, Feng Y-W, Lu C, Liu J-Q and Chen J-Y 2024 Vectorial generative adversarial surrogate modeling reliability evaluation framework for engineering structural systems *Reliab. Eng. Syst. Saf.* **247** 110076
[20] Wang R-Z, Gu H-H, Liu Y, Miura H, Zhang X-C and Tu S-T 2023 Surrogate-modeling-assisted creep-fatigue reliability assessment in a low-pressure turbine disc considering multi-source uncertainty *Reliab. Eng. Syst. Saf.* **240** 109550
[21] Lu C, Teng D, Chen J-Y, Fei C-W and Keshtegar B 2023 Adaptive vectorial surrogate modeling framework for multi-objective reliability estimation *Reliab. Eng. Syst. Saf.* **234** 109148
[22] Xu Y, Kohtz S, Boakye J, Gardoni P and Wang P 2023 Physics-informed machine learning for reliability and systems safety applications: state of the art and challenges *Reliab. Eng. Syst. Saf.* **230** 108900
[23] Fuhg J N, Fau A and Nackenhorst U 2021 State-of-the-art and comparative review of adaptive sampling methods for Kriging *Arch. Comput. Methods Eng.* **28** 2689–747
[24] Lataniotis C, Marelli S and Sudret B 2018 *Extending Classical Surrogate Modelling to Ultrahigh Dimensional Problems through Supervised Dimensionality Reduction: A Data-driven Approach* (ETH Zurich, Chair of Risk, Safety and Uncertainty Quantifi-cation)
[25] Zhao L, Wang P, Song B, Wang X and Dong H 2020 An efficient Kriging modeling method for high-dimensional design problems based on maximal information coefficient *Struct. Multidiscipl. Optim.* **61** 39–57
[26] Cheng G H, Younis A, Haji Hajikolaei K and Gary Wang G 2015 Trust region based mode pursuing sampling method for global optimization of high dimensional design problems *J. Mech. Des.* **137** 021407
[27] Bai F, Zou D and Wei Y 2024 A surrogate-assisted evolutionary algorithm with clustering-based sampling for high-dimensional expensive blackbox optimization *J. Glob. Optim.* **89** 93–115
[28] Zhao Y and Liu J 2025 An efficient solution method for inverse problems with high-dimensional model uncertainty parameters *Struct. Multidiscipl. Optim.* **68** 18
[29] Shan S and Wang G G 2010 Survey of modeling and optimization strategies to solve high-dimensional design problems with computationally-expensive black-box functions *Struct. Multidiscipl. Optim.* **41** 219–41
[30] Bouhlel M A, Bartoli N, Otsmane A and Morlier J 2016 Improving Kriging surrogates of high-dimensional design models by partial least squares dimension reduction *Struct. Multidiscipl. Optim.* **53** 935–52
[31] Chu H, Li W, Sun D, Liu Z, Zheng W, Xu J and Cheng Q 2025 AK-MCS structural reliability analysis method based on improved active learning function *J. Reliab. Sci. Eng.* **1** 025003
[32] Cao L and Zhao Y 2024 Uncertainty quantification for structural response field with ultra-high dimensions *Int. J. Mech. Sci.* **271** 109110
[33] Wu S, Hu J, Cao L, Huang K, Ma Y and Tang J 2026 Non-intrusive smoothed stochastic finite element method with multi-source uncertainties *Int. J. Mech. Sci.* **311** 111163
[34] Wang T, Li J, Lu D, Dong Y, Tan Y and Li Z 2024 A point mapping strategy-based sparse grid integration method for statistical moments estimation and structural reliability analysis *Comput. Methods Appl. Mech. Eng.* **430** 117238
[35] Shen S, Cheng J, Liu Z, Tan J and Zhang D 2025 Bayesian inference-assisted reliability analysis framework for robotic motion systems in future factories *Reliab. Eng. Syst. Saf.* **258** 110894
[36] Chen Y, Lu Z, Guo Y, Li X and Wu X 2025 Bayesian updating based method with dimension reduction integral technique for efficiently estimating multi-mode system failure probability function *Comput. Struct.* **316** 107859
[37] Guan X 2023 Moment quadrature method for uncertainty quantification of fatigue damage prognosis *Int. J. Fatigue* **172** 107654
[38] Guan X 2023 A moment quadrature method for uncertainty quantification of three-dimensional crack propagation via extremely few model runs *Comput. Methods Appl. Mech. Eng.* **411** 116061
[39] Guan X 2024 Sparse moment quadrature for uncertainty modeling and quantification *Reliab. Eng. Syst. Saf.* **241** 109665
[40] Gong T, He J and Guan X 2026 Continuously nested moment quadrature for uncertainty quantification of black-box models *Probab. Eng. Mech.* **82** 103892
[41] Mysovskikh I P 1968 On the construction of cubature formulas with the smallest number of nodes *Doklady*

Akademii Nauk (Russian Academy of Sciences)
[42] Golub G H and Welsch J H 1969 Calculation of Gauss quadrature rules *Math. Comput.* **23** 221–30
[43] Sanchez-Dehesa J 1978 The spectrum of Jacobi matrices in terms of its associated weight function *J. Comput. Appl. Math.* **4** 275–83
[44] De Branges L 1959 The Stone-Weierstrass theorem *Proc. Am. Math. Soc.* **10** 822–4
[45] He J, Gao S and Gong J 2014 A sparse grid stochastic collocation method for structural reliability analysis *Struct. Saf.* **51** 29–34
[46] Li P-P, Lu Z-H and Zhao Y-G 2022 An effective and efficient method for structural reliability considering the distributional parametric uncertainty *Appl. Math. Modell.* **106** 507–23
[47] Zech J and Schwab C 2020 Convergence rates of high dimensional Smolyak quadrature ESAIM *Math. Model. Numer. Anal.* **54** 1259–307
[48] Guan X 2025 The moment quadrature method for global sensitivity analysis *Reliab. Eng. Syst. Saf.* **265** 111460
[49] Saves P, Lafage R, Bartoli N, Diouane Y, Bussemaker J, Lefebvre T, Hwang J T, Morlier J and Martins J R 2024 SMT 2.0: a surrogate modeling toolbox with a focus on hierarchical and mixed variables Gaussian processes *Adv. Eng. Softw.* **188** 103571
[50] Barzilai J and Borwein J M 1988 Two-point step size gradient methods *IMA J. Numer. Anal.* **8** 141–8

Xuefei Guan is currently a Professor at Graduate School of China Academy of Engineering Physics. His research interests include ultrasonic nondestructive evaluation, fatigue and fracture, and uncertainty quantification methods applied to reliability engineering.