A gradient-enhanced univariate dimension reduction method for uncertainty propagation

Bingran Wang, Nicholas C. Orndorff, Mark Sperry, and John T. Hwang
University of California, San Diego, La Jolla, CA, USA

The univariate dimension reduction (UDR) method stands as a way to estimate the statistical moments of the output that is effective in a large class of uncertainty quantification (UQ) problems. UDR's fundamental strategy is to approximate the original function using univariate functions so that the UQ cost only scales linearly with the dimension of the problem. Nonetheless, UDR's effectiveness can diminish when uncertain inputs have high variance, particularly when assessing the output's second and higher-order statistical moments. This paper proposes a new method, gradient-enhanced univariate dimension reduction (GUDR), that enhances the accuracy of UDR by incorporating univariate gradient function terms into the UDR approximation function. Theoretical results indicate that the GUDR approximation is expected to be one order more accurate than UDR in approximating the original function, and it is expected to generate more accurate results in computing the output's second and higher-order statistical moments. Our proposed method uses a computational graph transformation strategy to efficiently evaluate the GUDR approximation function on tensor-grid quadrature inputs, and use the tensor-grid input-output data to compute the statistical moments of the output. With an efficient automatic differentiation method to compute the gradients, our method preserves UDR's linear scaling of computation time with problem dimension. Numerical results show that the GUDR is more accurate than UDR in estimating the standard deviation of the output and has a performance comparable to the method of moments using a third-order Taylor series expansion.

I. Introduction

Uncertainties are inherent in numerous scientific and engineering problems. Spanning from fields such as weather forecasting [1, 2], machine learning [3], structural analysis [4, 5] and aircraft design [6–8], these uncertainties can have a considerable effect on a system's behavior or performance. Forward uncertainty quantification (UQ), or uncertainty propagation, aims to evaluate how uncertainties in inputs influence the system's outputs or quantities of interest (QoIs). The input uncertainties can arise either from variations in operating conditions and model parameters, known as aleatoric uncertainties, or from model errors attributable to incomplete knowledge, known as epistemic uncertainties. By quantifying the impact of the input uncertainties, UQ provides critical support for decision-making and risk assessment in designing engineering systems. In this paper, we consider a UQ problem within a probabilistic framework, with the input uncertainties characterized using continuous probability density distributions. The objective here is to estimate the QoIs' statistical moments.

Developing computational methods to solve UQ problems has been a popular research topic for decades, and many approaches have been proposed to tackle UQ problems in different scenarios. The most common UQ methods include the method of moments, polynomial chaos, kriging, and Monte Carlo. The method of moments [9, 10] uses Taylor series approximations to compute statistical moments cost-effectively, but it may lack accuracy for large input variances and becomes impractical at higher expansion orders due to increased computational demand. Monte Carlo methods are commonly used for solving high-dimensional UQ problems as their convergence rate is unaffected by the dimensionality of the input space. Recent advancements seek to enhance its efficiency through multi-fidelity [11, 12]

and importance sampling techniques [13]. However, for low-dimensional scenarios, Monte Carlo may require an excessive number of model evaluations to match the precision of other UQ methods. Kriging, or Gaussian process regression, creates a surrogate response surface from input-output data, facilitating extensive model evaluations for reliability analysis [14, 15] or optimization under uncertainty [16]. The polynomial chaos approach models the QoI using orthogonal polynomials based on the distributions of the inputs, leveraging the smoothness in the random space for swift convergence with sampling or integration [17–19]. Both kriging and polynomial chaos are favored for low- to medium-dimensional problems but their computational cost does not scale well with the dimension of the problem, limiting their direct application in complex UQ tasks.

In tackling high-dimensional UQ problems, dimension reduction techniques offer a strategic means to curtail computational expenses. Among these, sensitivity analysis [20], partial least squares (PLS) [21], the active subspace (AS) method [22], and univariate dimension reduction (UDR) [23] are widely used. Sensitivity analysis, PLS, and AS leverage function or gradient evaluations at sampled input points to isolate key uncertain inputs or to uncover lower-dimensional latent variables in order to reduce the dimension of the problem. However, the success of these methods hinges on the existence of a compact set of latent variables that effectively capture the function's behavior. In contrast, UDR approximates the function as a sum of univariate functions, transforming a multidimensional integration problem into multiple one-dimensional problems. Consequently, the computational load of UDR increases only linearly with the number of dimensions, presenting a very cost-effective method for estimating the statistical moments of the QoIs. UDR has also been adapted for calculating complete probability distributions [24] and reliability metrics of QoIs [25, 26]. Nevertheless, its precision diminishes with increasing input variances, particularly when estimating second-order and higher statistical moments. To enhance UDR's accuracy, some extensions incorporate bivariate functions into the UDR approximation [27]. Although these adjustments provide better accuracy, they compromise UDR's linear scalability, presenting a trade-off that requires careful consideration in the method's application.

Wang et al. recently introduced a novel method known as *Accelerated Model evaluations on Tensor grids using Computational graph transformations* (AMTC) [28, 29]. This method reduces the model evaluation cost on tensor-grid inputs by modifying the computational graph of the model. This modification eliminates the redundant evaluations on the operation level that are incurred by the tensor structure of the inputs. The existing work applies AMTC to integration-based non-intrusive polynomial chaos, assuming a certain type of sparsity is present in the computational graph of the original computational model. In some problems, this sparsity enables speedups of multiple orders of magnitude [30–32]. However, if the computational graph does not possess this type of sparsity, the AMTC-based UQ methods may be outperformed by other UQ methods.

This paper proposes a new UQ method, named gradient-enhanced univariate dimension reduction (GUDR), intending to enhance the accuracy of UDR by incorporating univariate gradient functions into the approximations. The GUDR approximation utilizes univariate function evaluations, univariate gradient evaluations, and a single Hessian evaluation at the mean values of the inputs. Theoretical results show that the GUDR approximation is one order more accurate in estimating the original function, and is expected to have a comparable level of accuracy as the third-order Taylor series expansion in estimating the second and higher-order statistical moments of the output. To estimate the statistical moments of the output, we propose to use the AMTC strategy to effectively perform the evaluations of the GUDR approximation function on tensor-grid quadrature points. This tensor-grid input-output data can also be used to compute the risk measures of the output. The results show that, with an efficient automatic differentiation method, the computational cost of the GUDR method scales linearly with the dimension of the problem.

This method has been applied to five UQ problems: one 2D and one 3D mathematical function, one mathematical function with varying dimensions, one 4D rotor aerodynamic analysis problem, and one 7D problem derived from a practical aircraft design scenario. The results indicate that on the first two mathematical functions, when estimating the standard deviation of the output, GUDR is more accurate than UDR and has comparable performance as the method of moments method using third-order Taylor expansion. On the third mathematical function, numerical results show that the function evaluation cost of GUDR scales linearly with the problem dimension and is less than four times that of UDR. On the 4D rotor analysis and 7D aircraft design problems, GUDR enhances the accuracy of UDR by an order of magnitude in estimating the standard deviation and becomes the most cost-effective method to use among the UQ methods implemented.

This paper is organized as follows. Section II provides some background for UDR and AMTC. Section III presents the details of the GUDR method. Section IV shows numerical results on the test problems. Section V summarizes the work and offers concluding thoughts.

II. Background

A. Univariate dimension reduction

We consider an uncertainty quantification (UQ) problem involving a function $f(u)$, where $u \in \mathbb{R}^d$ denotes the input vector, and $f \in \mathbb{R}$ represents a scalar output. The uncertain inputs are expressed as a stochastic vector denoted by $U := [U_1, \ldots, U_d]^T$, with the assumption that these random variables are mutually independent. The stochastic input vector has probability density distribution $\rho(u)$ with support Γ. The UQ problem aims to compute the statistical moments of the output random variable, $f(U)$. Given that the mean of the uncertain inputs vector is $\mu := [\mu_1, \ldots, \mu_d]^T$, the univariate dimension reduction (UDR) [23] method approximates the original function $f(u)$ using a sum of univariate functions:

$$\hat{f}(u) = \sum_{i=1}^{d} f_i(u_i) - (d-1)f(\mu), \tag{1}$$

where each univariate function term is defined as $f_i(u_i) := f(\mu_1, \ldots, \mu_{i-1}, u_i, \mu_{i+1}, \ldots, \mu_d)$. Expanding $f(u)$ in a Taylor series at $u = \mu$ yields

$$f(u) = f(\mu) + \sum_{j=1}^{\infty} \sum_{i=1}^{d} \frac{1}{j!} \frac{\partial^j f}{\partial u_i^j}(\mu)(u_i - \mu_i)^j + \sum_{j_2=1}^{\infty} \sum_{j_1=1}^{\infty} \frac{1}{j_1! j_2!} \sum_{i_1=1}^{d-1} \sum_{i_2 > i_1}^{d} \frac{\partial^{j_1 + j_2} f}{\partial u_{i_1}^{j_1} \partial u_{i_2}^{j_2}}(\mu)(u_{i_1} - \mu_{i_1})^{j_1}(u_{i_2} - \mu_{i_2})^{j_2}$$

$$+ \sum_{j_3=1}^{\infty} \sum_{j_2=1}^{\infty} \sum_{j_1=1}^{\infty} \frac{1}{j_1! j_2! j_3!} \sum_{i_1=1}^{d-2} \sum_{i_2 > i_1}^{d-1} \sum_{i_3 > i_2}^{d} \frac{\partial^{j_1 + j_2 + j_3} f}{\partial u_{i_1}^{j_1} \partial u_{i_2}^{j_2} \partial u_{i_3}^{j_3}}(\mu)(u_{i_1} - \mu_{i_1})^{j_1}(u_{i_2} - \mu_{i_2})^{j_2}(u_{i_3} - \mu_{i_3})^{j_3} + \ldots \tag{2}$$

The Taylor series expansion of the UDR approximation function, $\hat{f}(u)$, at $u = \mu$ can be expressed as

$$\hat{f}(u) = f(\mu) + \sum_{j=1}^{\infty} \sum_{i=1}^{d} \frac{1}{j!} \frac{\partial^j f}{\partial u_i^j}(\mu)(u_i - \mu_i)^j. \tag{3}$$

If we compare the Taylor series of the original function with the UDR approximation function, all of the Taylor series terms in (3) are contained in (2), and the residual errors are

$$f(u) - \hat{f}(u) = \sum_{j_2=1}^{\infty} \sum_{j_1=1}^{\infty} \frac{1}{j_1! j_2!} \sum_{i_1=1}^{d-1} \sum_{i_2 > i_1}^{d} \frac{\partial^{j_1 + j_2} f}{\partial u_{i_1}^{j_1} \partial u_{i_2}^{j_2}}(\mu)(u_{i_1} - \mu_{i_1})^{j_1}(u_{i_2} - \mu_{i_2})^{j_2}$$

$$+ \sum_{j_3=1}^{\infty} \sum_{j_2=1}^{\infty} \sum_{j_1=1}^{\infty} \frac{1}{j_1! j_2! j_3!} \sum_{i_1=1}^{d-2} \sum_{i_2 > i_1}^{d-1} \sum_{i_3 > i_2}^{d} \frac{\partial^{j_1 + j_2 + j_3} f}{\partial u_{i_1}^{j_1} \partial u_{i_2}^{j_2} \partial u_{i_3}^{j_3}}(\mu)(u_{i_1} - \mu_{i_1})^{j_1}(u_{i_2} - \mu_{i_2})^{j_2}(u_{i_3} - \mu_{i_3})^{j_3} + \ldots \tag{4}$$

This shows that the UDR approximation is only second-order accurate in approximating the original function [23]. However, when it comes to estimating the mean of the output, the residual errors become

$$\mathbb{E}[f(U)] - \mathbb{E}[\hat{f}(U)] = \int_{\Gamma_1} \ldots \int_{\Gamma_d} \left(f(u) - \hat{f}(u) \right) \rho(u) du_1 \ldots du_d$$

$$= \frac{1}{2! 2!} \sum_{i_1=1}^{d-1} \sum_{i_2 > i_1}^{d} \frac{\partial^4 f}{\partial u_{i_1}^2 \partial u_{i_2}^2}(\mu) \mathbb{E}[(u_{i_1} - \mu_{i_1})^2 (u_{i_2} - \mu_{i_2})^2] + \ldots, \tag{5}$$

which involves only fourth and higher-order integration terms. In contrast, the 3rd-order Taylor series method approximates the function as

$$\tilde{f}(u) = f(\mu) + \sum_{j=1}^{3} \sum_{i=1}^{d} \frac{1}{j!} \frac{\partial^j f}{\partial u_i^j}(\mu)(u_i - \mu_i)^j + \sum_{j_1=1}^{2} \sum_{j_2=1}^{j_1 + j_2 \leq 3} \frac{1}{j_1! j_2!} \sum_{i_1=1}^{d-1} \sum_{i_2 > i_1}^{d} \frac{\partial^{j_1 + j_2} f}{\partial u_{i_1}^{j_1} \partial u_{i_2}^{j_2}}(\mu)(u_{i_1} - \mu_{i_1})^{j_1}(u_{i_2} - \mu_{i_2})^{j_2}$$

$$+ \sum_{i_1=1}^{d-2} \sum_{i_2 > i_1}^{d-1} \sum_{i_3 > i_2}^{d} \frac{\partial^3 f}{\partial u_{i_1} \partial u_{i_2} \partial u_{i_3}}(\mu)(u_{i_1} - \mu_1)(u_{i_2} - \mu_2)(u_{i_3} - \mu_3). \tag{6}$$

3

When estimating the mean of the output with this method, the residual errors are

$$
\mathbb{E}[f(U)] - \mathbb{E}[\tilde{f}(U)] = \frac{1}{2!2!} \sum_{i_1=1}^{d-1} \sum_{i_2>i_1}^{d} \frac{\partial^4 f}{\partial u_{i_1}^2 \partial u_{i_2}^2}(\mu) \mathbb{E}[(u_{i_1} - \mu_{i_1})^2 (u_{i_2} - \mu_{i_2})^2]
$$
$$
+ \sum_{i=1}^{d} \frac{1}{4!} \frac{\partial^4 f}{\partial u_i^4}(\mu) \mathbb{E}[(u_i - \mu_i)^4] + \ldots \tag{7}
$$

Upon comparing the UDR's residual errors in (5) with those of the 3rd-order Taylor series in (7), we observe that both methods are fourth-order accurate in estimating the mean of the output. However, the UDR comprises fewer fourth-order integration terms. If we assume the constants on the fourth-order terms are similar, it suggests that UDR generally yields a more accurate result than the 3rd-order Taylor expansion when estimating the mean of the output.

Nevertheless, for higher-order statistical moments, the UDR approximation may not perform as well as the 3rd-order Taylor expansion. For instance, when estimating the second-order statistical moments, the UDR's relative errors are:

$$
\mathbb{E}[f(U)^2] - \mathbb{E}[\hat{f}(U)^2] = \sum_{i_1=1}^{d-1} \sum_{i_2>i_1}^{d} \left(\frac{\partial^2 f}{\partial u_{i_1} \partial u_{i_2}}(\mu) \right)^2 \mathbb{E}[(u_{i_1} - \mu_{i_1})^2 (u_{i_2} - \mu_{i_2})^2]
$$
$$
+ \sum_{i_1=1}^{d-1} \sum_{i_2>i_1}^{d} \left(\frac{\partial f}{\partial u_{i_1}}(\mu) \frac{\partial^3 f}{\partial u_{i_1} \partial u_{i_2}^2}(\mu) + \frac{\partial f}{\partial u_{i_2}}(\mu) \frac{\partial^3 f}{\partial u_{i_1}^2 \partial u_{i_2}}(\mu) \right) \mathbb{E}[(u_{i_1} - \mu_{i_1})^2 (u_{i_2} - \mu_{i_2})^2] \tag{8}
$$
$$
+ \frac{1}{2!} f(\mu) \sum_{i_1=1}^{d-1} \sum_{i_2>i_1}^{d} \frac{\partial^4 f}{\partial u_{i_1}^2 \partial u_{i_2}^2}(\mu) \mathbb{E}[(u_{i_1} - \mu_{i_1})^2 (u_{i_2} - \mu_{i_2})^2] + \ldots
$$

In comparison, the residual errors for the 3rd order Taylor series method are

$$
\mathbb{E}[f(U)^2] - \mathbb{E}[\tilde{f}(U)^2] = \frac{1}{2!} f(\mu) \sum_{i_1=1}^{d-1} \sum_{i_2>i_1}^{d} \frac{\partial^4 f}{\partial u_{i_1}^2 \partial u_{i_2}^2}(\mu) \mathbb{E}[(u_{i_1} - \mu_{i_1})^2 (u_{i_2} - \mu_{i_2})^2] + \ldots \tag{9}
$$

Comparing the residual errors of the 3rd-order Taylor series method in (9) with the residual errors of the UDR in (8), both of the methods are also fourth-order accurate in estimating the second-order moment of the output, but the relative errors of the UDR comprise significantly more fourth-order integration terms compared with the 3rd-order Taylor series expansion. Thus it is expected that the UDR is not as accurate as the 3rd Taylor series expansion method when used to estimate the second order moment of the output.

The detailed derivation of the residual errors for the UDR approximation in estimating the first and second-order statistical moments for a 2D problem can be found in Appendix V.A.
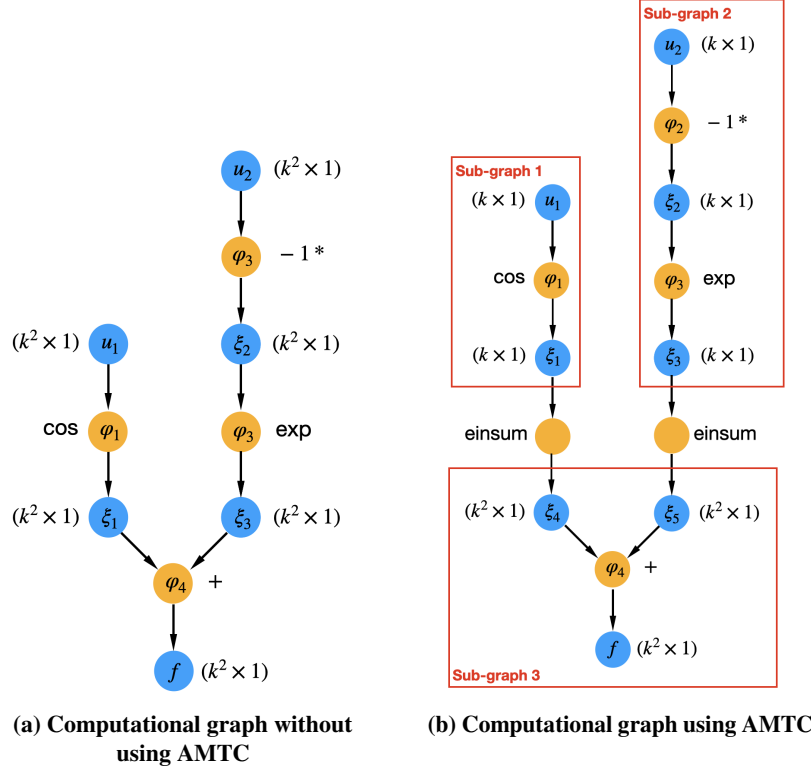
1. Computational cost

The major advantage of the UDR approximation is that when estimating the statistical moments of the output, the corresponding multidimensional integration problem can be decomposed into multiple one-dimensional integration problems, which significantly reduces the required computational cost. For example, when computing the mean of the output, the UDR yields

$$
\mathbb{E}[\hat{f}(U)] = \sum_{i=1}^{d} \int_{\Gamma_i} f_i(u_i)\rho(u_i)du_i - (d-1)f(\mu). \tag{10}
$$

This only requires performing d one-dimensional integrations instead of performing a d-dimensional integration. If we use the quadrature rule to approximate each integral with k quadrature points, the total number of model evaluations required is $kd + 1$, which scales only linearly with the dimension of the UQ problem. For higher-order statistical moments of the output, UDR can compute it following a recurring formula in [23] without requiring extra model evaluations.

B. Computational graph transformations to accelerate tensor-grid evaluations

Wang et al. recently proposed the *Accelerated Model evaluations on Tensor grids using Computational graph transformations* (AMTC) method in [28]. AMTC is a graph transformation method to minimize the computational

(a) Computational graph without using AMTC

(b) Computational graph using AMTC

Fig. 1 **Computational graphs with data size for full-grid input points evaluation on** $f = cos(u_1) + exp(-u_2)$ **[28]**

cost required to evaluate a computational model on tensor-grid input points. It represents the computational model as a computational graph built from basic operations and capitalizes on the fact that each operation only needs to be evaluated on the unique points in its input space. When we evaluate a computational model at tensor-grid input points, many operations depend on a small subset of the uncertain inputs. This leads to redundant evaluations at the operation level with a naive approach.

The AMTC method removes redundant calculations by partitioning the computational graph into smaller sub-graphs, based on the dependency information of the operations. Each sub-graph is computed just once based on the dependency information of the operations. Operations in each graph share the same input space and are evaluated only on the same distinct nodes in their input space. To maintain proper data flow among these sub-graphs, Einstein summation nodes are introduced to connect the sub-graphs. For illustration, take the function, $f = \cos(u_1) + \exp(-u_2)$ evaluated at full-grid input points where each dimension has k points. This function is broken down into fundamental operations:

$$
\begin{aligned}
\xi_1 &= \varphi_1(u_1) = \cos(u_1); \\
\xi_2 &= \varphi_2(u_2) = -u_2; \\
\xi_3 &= \varphi_3(\xi_2) = \exp(\xi_2); \\
f &= \varphi_4(\xi_1, \xi_3) = \xi_1 + \xi_3.
\end{aligned}
\tag{11}
$$

The tensor-grid input points are given as

$$
\boldsymbol{u} = \left\{
\begin{matrix}
(u_1^{(1)}, u_2^{(1)}) & \cdots & (u_1^{(k)}, u_2^{(1)}) \\
\vdots & \ddots & \vdots \\
(u_1^{(1)}, u_2^{(k)}) & \cdots & (u_1^{(k)}, u_2^{(k)})
\end{matrix}
\right\},
\tag{12}
$$

with k^2 input points. The computational graphs to evaluate this function with and without the AMTC method are shown in Fig. 1. Comparing the computational graphs reveals that without AMTC, every operation in the graph is evaluated k^2 times. But by applying AMTC, operations that depend solely on u_1 or u_2 are calculated just k times, recognizing

5

that their outputs have k unique values. This method is especially beneficial when combined with the non-intrusive polynomial chaos method or quadrature rule for solving UQ problems when the model's computational graph is not densely connected. AMTC has been integrated into the compiler's middle end for the *Computational System Design Language* (CSDL)[33], a newly developed domain-specific language tailored for the solution of multidisciplinary design, analysis, and optimization problems. Within the CSDL compiler, the AMTC serves as a middle-end algorithm, generating the modified computational graph to minimize the model evaluation cost on tensor-grid inputs.

III. Methodology

A. Gradient-enhanced univariate dimension reduction

We now describe the new methodology that forms the primary contribution of this paper. The key idea here is to postulate a new approximation expression that enhances the accuracy of the univariate dimension reduction (UDR) approximation expression by adding the univariate gradient terms. With the addition of these gradient terms, the new approximation expression can be more accurate in estimating the second and higher-order statistical moments while the required computational cost still scales linearly with the problem dimension. We coin the new method as the *gradient-enhanced univariate dimension reduction* (GUDR) method. GUDR approximates the original function, f, as

$$
\begin{aligned}
\bar{f}(u) = &\sum_{i=1}^{d} f_i(u_i) - (d-1)f(\mu) + \sum_{i=1}^{d}(u_i - \mu_i)\sum_{j\neq i}^{d} \frac{\partial f_j}{\partial u_i}(u_j) \\
&- (d-1)\sum_{i}^{d}(u_i - \mu_i)\frac{\partial f}{\partial u_i}(\mu) - \sum_{i=1}^{d}\sum_{j>i}^{d}\frac{\partial^2 f}{\partial u_i \partial u_j}(\mu)(u_i - \mu_i)(u_j - \mu_j),
\end{aligned}
\tag{13}
$$

where $\frac{\partial f_j}{\partial u_i}(u_j) := \frac{\partial f}{\partial u_i}(\mu_1, \ldots, \mu_{j-1}, u_j, \mu_{j+1}, \ldots, \mu_d)$ represents a univariate gradient term. This form includes all of the terms in the UDR approximation function in (1), with additional terms including the univariate first-order gradient terms and the second-order derivatives evaluated at μ. The GUDR approximation can also be written in matrix form as

$$
\begin{aligned}
\bar{f}(x) = &\sum_{i=1}^{d} f_i(u_i) - (d-1)f(\mu) + \sum_{i=1}^{d} \underbrace{\begin{bmatrix} u_1 - \mu_1 \\ \vdots \\ u_{i-1} - \mu_{i-1} \\ 0 \\ u_{i+1} - \mu_{i+1} \\ \vdots \\ u_d - \mu_d \end{bmatrix}^T}_{(u-\mu)\odot(\mathbf{1}-\mathbf{e_i})} \underbrace{\begin{bmatrix} \frac{\partial f_i}{\partial u_1}(u_i) \\ \vdots \\ \frac{\partial f_i}{\partial u_d}(u_i) \end{bmatrix}}_{\frac{\partial f_i}{\partial u}(u_i)} \\
&- (d-1)\underbrace{\begin{bmatrix} u_1 - \mu_1 \\ \vdots \\ u_d - \mu_d \end{bmatrix}^T}_{u-\mu} \underbrace{\begin{bmatrix} \frac{\partial f}{\partial u_1}(\mu) \\ \vdots \\ \frac{\partial f}{\partial u_d}(\mu) \end{bmatrix}}_{\frac{\partial f}{\partial u}(\mu)} - \underbrace{\begin{bmatrix} u_1 - \mu_1 \\ \vdots \\ u_d - \mu_d \end{bmatrix}^T \begin{bmatrix} 0 & \frac{\partial^2 f}{\partial u_1 \partial u_2}(\mu) & \cdots & \frac{\partial^2 f}{\partial u_1 \partial u_d}(\mu) \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & \frac{\partial^2 f}{\partial u_{d-1}\partial u_d}(\mu) \\ 0 & \cdots & \cdots & 0 \end{bmatrix} \begin{bmatrix} u_1 - \mu_1 \\ \vdots \\ u_d - \mu_d \end{bmatrix}}_{\frac{1}{2}(u-\mu)^T\left(\frac{\partial^2 f}{\partial u^2}(\mu) - \mathrm{diag}\left(\mathrm{diag}\left(\frac{\partial^2 f}{\partial u^2}(\mu)\right)\right)\right)(u-\mu)} \\
= &\sum_{i=1}^{d} f_i(u_i) + \sum_{i=1}^{d}\left((u-\mu)\odot(\mathbf{1}-\mathbf{e_i})\right)^T \frac{\partial f_i}{\partial u}(u_i) - (d-1)\left(f(\mu) - (u-\mu)^T\frac{\partial f}{\partial u}(\mu)\right) \\
&- \frac{1}{2}(u-\mu)^T\left(\frac{\partial^2 f}{\partial u^2}(\mu) - \mathrm{diag}\left(\mathrm{diag}\left(\frac{\partial^2 f}{\partial u^2}(\mu)\right)\right)\right)(u-\mu),
\end{aligned}
\tag{14}
$$

6

where $\frac{\partial f}{\partial u}(\mu)$ and $\frac{\partial^2 f}{\partial u^2}(\mu)$ represent the gradient vector and Hessian matrix evaluated at $u = \mu$, respectively. If we compare the Taylor series expansions at $u = \mu$ of the GUDR approximation function and the original function, just as with the UDR approximation, all of the Taylor series terms of the GUDR approximation function are included in the original function, and the residual errors can be expressed as

$$f(u) - \bar{f}(u) = \sum_{i=1}^{d} \sum_{j>i}^{d} \sum_{k>j}^{d} \frac{\partial^3 f}{\partial u_i \partial u_j \partial u_k}(\mu)(u_i - \mu_i)(u_j - \mu_j)(u_k - \mu_k) + \dots . \tag{15}$$

This shows the GUDR approximation is third-order accurate when used to approximate the original function. In comparison with the residual errors of the UDR approximation in (3), the GUDR approximation is a more accurate approximation of the original function as the UDR approximation is only second-order accurate.

When it comes to estimating the mean of the output, the GUDR approximation generates the same results as the UDR approximation with the residual errors expressed as

$$\mathbb{E}[f(U)] - \mathbb{E}[\bar{f}(U)] = \frac{1}{2!2!} \sum_{i_1=1}^{d-1} \sum_{i_2>i_1}^{d} \frac{\partial^4 f}{\partial u_{i_1}^2 \partial u_{i_2}^2}(\mu)\mathbb{E}[(u_{i_1} - \mu_{i_1})^2(u_{i_2} - \mu_{i_2})^2] + \dots . \tag{16}$$

This means that the UDR and GUDR approximations are expected to provide the same results in estimating the mean of the output, and both are expected to be more accurate than the 3rd-order Taylor series expansion. However, when it comes to estimating the higher-order statistical moments, the GUDR can provide more accurate results than UDR as the GUDR approximation is more accurate than the UDR approximation in estimating the original function. For example, when estimating the second-order statistical moment of the output, the residual errors of the GUDR approximation can be expressed as

$$\mathbb{E}[f(U)^2] - \mathbb{E}[\bar{f}(U)^2] = \frac{1}{2!}f(\mu) \sum_{i_1=1}^{d-1} \sum_{i_2>i_1}^{d} \frac{\partial^4 f}{\partial u_{i_1}^2 \partial u_{i_2}^2}(\mu)\mathbb{E}[(u_{i_1} - \mu_{i_1})^2(u_{i_2} - \mu_{i_2})^2] + \dots \tag{17}$$

When compared with the residual errors from the 3rd-order Taylor series expansion method, as detailed in (9), and those from the UDR method, seen in (8), all three methods exhibit residual errors that include only terms of the fourth order and higher. However, the residual errors of the GUDR and 3rd Taylor series expansion approximations contain the same fourth-order integration terms, while the UDR contains significantly more fourth-order integration terms. This means that, when estimating the second-order statistical moments, the GUDR approximation is expected to have a comparable level of accuracy with the 3rd-order Taylor series expansion, and its result can be significantly more accurate than the UDR approximation.

The detailed derivation of the residual errors for the GUDR approximation in estimating the first and second-order statistical moments for a 2D problem can be found in Appendix V.B.

B. Tensor-grid evaluations to estimate statistical moments

For a d-dimensional UQ problem, the GUDR approximation in (14) involves multiple univariate function terms and univariate gradient terms. When it comes to estimating the mean of the output, all of the gradient terms become zero, and the GUDR yields the same result as UDR,

$$\mathbb{E}[\bar{f}(U)] = \sum_{i=1}^{d} \int_{\Gamma_i} f_i(u_i)\rho(u_i)du_i - (d-1)f(\mu), \tag{18}$$

which can be evaluated following the UDR method. However, when it comes to estimating higher-order statistical moments of the output, there is no easy way to decompose the multidimensional integrals involved. One may want to derive the recursive formula for GUDR following the UDR method. However, the recursive formula will be significantly more complicated and is unpractical to implement in real problems.

In this paper, we propose a new approach to decompose the GUDR approximation function from the computational-graph-transformation perspective, so that we efficiently generate the model evaluations on full-grid quadrature points. This approach is inspired by the recently developed computational graph transformation method, *Accelerated Model evaluations on Tensor-grid using Computational graph transformation* (AMTC).

7

If we treat each univariate function and gradient term in the GUDR approximation as individual operations, the computational graph of the GUDR approximation function is shown in Fig. 2. Despite the complicated form of the GUDR approximation, the main computational cost only comes from evaluating the univariate function and gradient terms. When applying the GUDR approximation function to tensor-grid inputs, which are described as

$$\boldsymbol{u} = \boldsymbol{u}_1^k \times \ldots \times \boldsymbol{u}_d^k, \tag{19}$$

where $\boldsymbol{u}_i^k$ denotes the set of k quadrature points within the u_i dimension, we note that despite the presence of k^d input points, only k distinct points exist within each input dimension. Drawing on AMTC's foundational concept, since every univariate function and gradient computation operation in the GUDR's computational graph is dependent on just one uncertain input, it is only necessary to evaluate these operations at the distinct quadrature points within their input space.

Consequently, by integrating the AMTC strategy, the modified computational graph for the GUDR approximation function's tensor-grid evaluations is shown in Fig. 3. In this way, we can generate the model evaluations of GUDR approximation function on tensor-grid inputs, $\bar{f}(\boldsymbol{u}) \in \mathcal{R}^{k^d}$, with only k evaluations on each univariate function and gradient evaluation term.

We note that the AMTC method has only been implemented in the CSDL compiler and can only be applied to computational models that are built in the CSDL language. Fortunately, in this specific scenario, we can achieve the minimum number of evaluations in univariate function and gradient evaluation terms by manually adding the *Einsum** operations. First, we evaluate the univariate model function and gradient function on the corresponding quadrature points in its input space. Then we manually add *Einsum* operations in the code to transform these quadrature points evaluations to the correct size with the correct order of the data. Lastly, this data is passed to the GUDR approximation function to generate the full-grid evaluations of the output. We show example codes to achieve this in the Appendix V.C.

The tensor-grid input-output data can be easily used with the quadrature rule to compute any order of statistical moments of the output. Additionally, this data can also be used with other UQ methods like non-intrusive polynomial chaos, stochastic collocation, and kriging to construct a surrogate model to approximate the risk measures of the output, such as the probability of failure or conditional value at risk.

C. Computational cost

Utilizing the AMTC strategy enables the computation of any statistical moment or risk measure of the output with a fixed number of evaluations. Specifically, GUDR would only require k evaluations for each univariate function and its gradient term, accompanied by a single function evaluation, gradient evaluation, and Hessian evaluation at the mean of the input variables. With reverse-mode automatic differentiation, for a function with d inputs, a single gradient vector evaluation, and a full Hessian matrix evaluation carry the computational weight less than 3 and $3d$ model evaluations [34], respectively. As a result, a conservative estimation of the model evaluation cost for solving the d-dimensional UQ using GUDR with k quadrature points in each dimension of the uncertain input is no more than $4kd + 3d + 4$ model evaluations, which scales only linearly with the dimension of the problem.

IV. Numerical Results

In Sec IV.A, we compare the accuracy of four UQ methods for estimating the standard deviation of the output: gradient-enhanced univariate dimension reduction (GUDR), univariate dimension reduction (UDR), method of moments using a 2nd order Taylor series expansion, and method of moments using a 3rd order Taylor series expansion. These methods are applied to two UQ problems involving different mathematical functions. In Sec IV.A, we compare the scalability of three UQ methods: UDR, GUDR, and full-grid quadrature rule. These methods are applied to one UQ problem involving a mathematical function with varying dimensions. In Sec IV.C and Sec IV.D, we compare the convergence plots of GUDR, UDR, and the Monte Carlo methods in estimating both the mean and the standard deviation of the output on a 4D UQ problem involving a rotor aerodynamic analysis model and a 7D UQ problem involving an aircraft design multidisciplinary model, respectively.

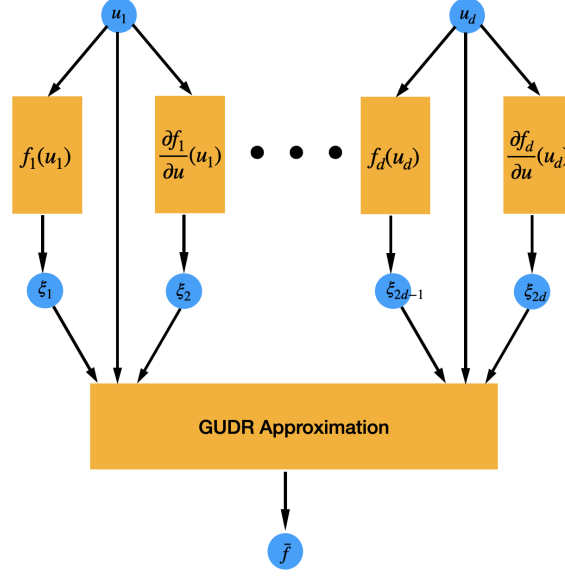*Einsum refers to the Einstein summation function in Numpy.

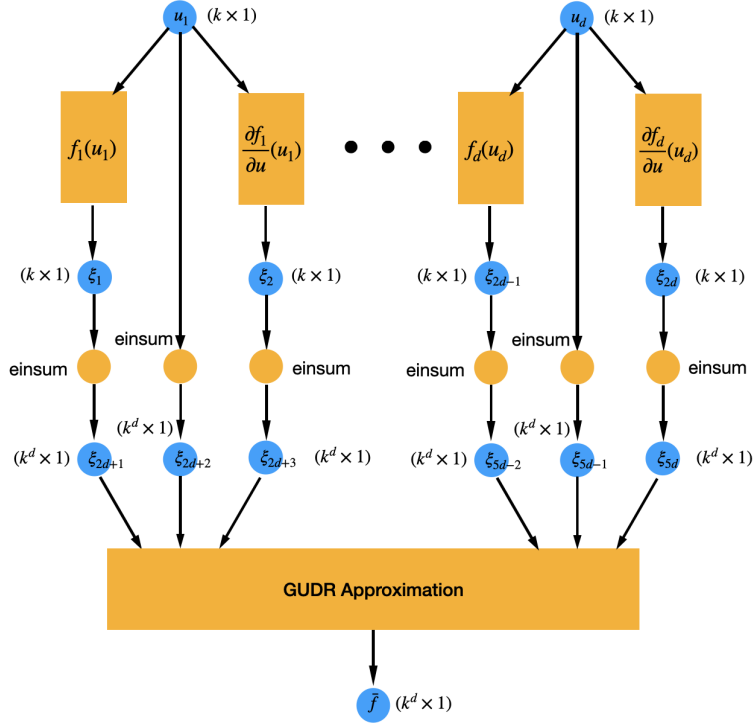**Fig. 2   Computational graph of the GUDR approximation function**

**Fig. 3   Computational graph of tensor-grid evaluations after graph transformation**

A. Mathematical functions with varying input variances

We first examine two UQ problems that involve mathematical functions defined by simple, closed-form expressions. The first problem involves the function,

$$y_1 = f(x_1, x_2) = \frac{1}{1 + x_1^4 + 2x_2^2 + x_2^4} \tag{20}$$
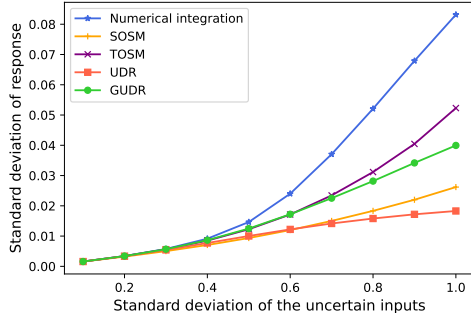
(a) QoI results for increasing values of input standard deviation
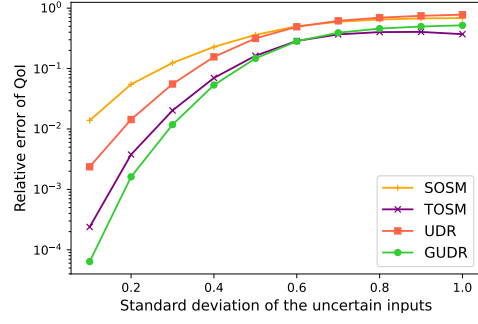
(b) Relative errors for increasing values of input standard deviation

Fig. 4   UQ results for $y_1 = (1 + x_1^4 + 2x_2^2 + 5_2^4)^{-1}$

with uncertain inputs,

$$X_i \sim \mathcal{N}(2, \sigma), \quad i = 1, 2. \tag{21}$$

The second problem involves the function:

$$y_2 = f(x_1, x_2, x_3) = \exp(1 + 0.5x_1^2 + 0.5x_2^2 + 0.5x_3^2) \tag{22}$$

with uncertain inputs:

$$X_i \sim \mathcal{N}(3, \sigma), \quad i = 1, 2, 3. \tag{23}$$

The quantity of interest (QoI) for both UQ problems is the standard deviation of the output Y_1 and Y_2, respectively. These problems are adapted from [23].

For these test problems, we have implemented five methods: UDR, GUDR, method of moments using the 2nd-order Taylor series expansion (second-order second-moment (SOSM)), method of moments using the 3rd-order Taylor series expansion (third-order second-moment (TOSM)), and direct numerical integration. The UDR and GUDR methods are implemented using 19 quadrature points in each dimension of the uncertain inputs to yield the most accurate UQ results based on these two analytical approximation expressions. Conversely, the direct numerical integration method utilizes the Monte Carlo method with 100,000 sample points, with its UQ results considered as the ground truth.

The UQ results derived from these five methods, along with the relative errors of the four analytical expression methods, are plotted against the increasing values of input standard deviations in Fig. 4 and Fig. 5 for Y_1 and Y_2, respectively. Our observations indicate that for both problems, all of the UDR, GUDR, SOSM, and TOSM methods yield accurate results for the standard deviation of the response when the standard deviation of the uncertain inputs is small. As we increase the standard deviation, the errors for all four methods correspondingly escalate. While the performances of UDR and SOSM are comparable, GUDR and TOSM also have comparable performances and consistently outperform UDR and SOSM. Typically, the relative errors of GUDR are less than those of UDR and SOSM by more than an order of magnitude. This superior performance of GUDR can be attributed to its more accurate approximation of the original function in comparison to UDR and the 2nd-order Taylor expansion. Additionally, the comparable performances of GUDR and TOSM match our theoretical results showing that the GUDR approximation function is expected to have comparable performance as the 3rd-order Taylor expansion when estimating the standard deviation of the output.

B. Mathematical function with varying dimensions

The third test problem involves an analytical problem with varying dimension:

$$y_3 = f(x_1, \ldots, x_d) = \exp\left(1 + \sum_{i=1}^{d} 0.5x_i^2\right) \tag{24}$$

with uncertain inputs:

$$X_i \sim \mathcal{N}(3, 1), \quad i = 1, \ldots, d. \tag{25}$$

10

(a) QoI results for increasing values of input standard deviation

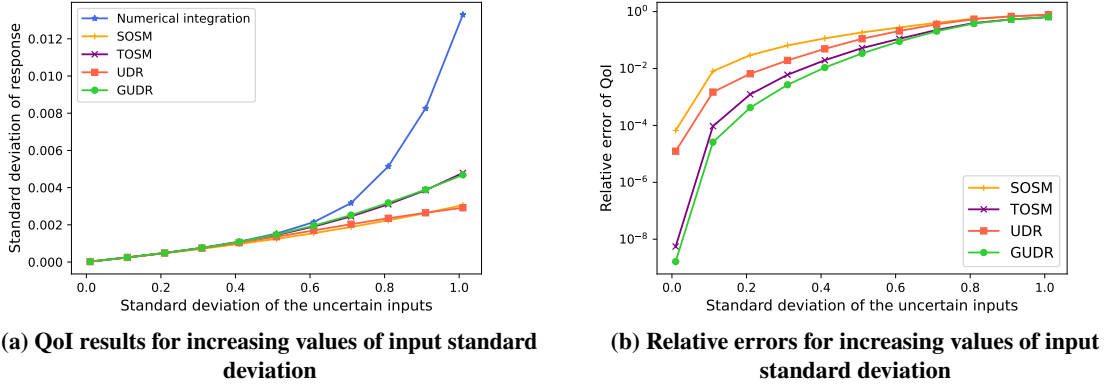**(b) Relative errors for increasing values of input standard deviation**

Fig. 5   UQ results for $y_2 = e^{(1+0.5x_1^2+0.5x_2^2+0.5x_3^2)}$

For this test problem, we aim to investigate how the function evaluation cost scales with the dimension of the function for the GUDR method. We have implemented three methods: UDR, GUDR and direct numerical integration with full-grid quadrature rule. For all three methods, we use $k = 5$ as the number of quadrature points in each dimension. For UDR and GUDR, the derivatives are computed using the automatic differentiation from TensorFlow. The function evaluations time for each case is measured as the average of 200 runs.

The scalability results are plotted inFig. 6. Our observations indicate that both UDR and GUDR show linear scalability with the problem dimension, while numerical integration with the full-grid quadrature rule exhibits exponential scalability. Comparing GUDR with UDR, the results show that the computational cost of GUDR is consistently larger than that of UDR, with a ratio between 2 and 4. These observations align with our theoretical analysis. When using an efficient automatic differentiation method to compute the gradients, both GUDR and UDR scale linearly with the problem dimension, and the computational cost of GUDR is estimated to be no more than four times that of UDR.
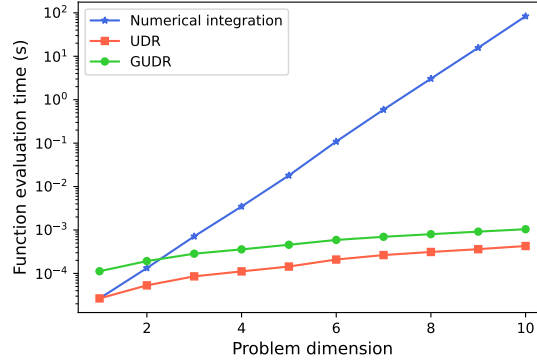

Fig. 6   Scalability plots for $y_3 = \exp(1 + \sum_{i=1}^{d} 0.5x_i^2)$

C. Rotor aerodynamic analysis model

The fourth test problem is a 4D UQ problem involving the aerodynamic analysis of a rotor on an aircraft. The computational model applies the blade element momentum (BEM) theory to compute the rotor performance metrics including the thrust and torque generated. The streamtube analyzed by the BEM model is shown in Fig. 7. Further details of the model can be found in [35]. The objective of the UQ problem is to determine the mean and standard deviation of the generated torque, given four uniform uncertain inputs shown in Tab. 1.

In this and the following test problems, we implemented four UQ methods: GUDR, UDR, non-intrusive polynomial chaos (NIPC), and the Monte Carlo method. The computational models were implemented in the Computational System

11

Design Language (CSDL) [33]. The first-order gradient evaluations required for GUDR were calculated using an automatically implemented adjoint method detailed in [36], and the Hessian matrix was estimated via a finite difference method. The NIPC results were generated using the software package Chaospy [37], using a regression-based approach. We use the UQ results generated from the Monte Carlo method with $10,000$ sample points as the reference results and depict the convergence plots of the four methods in Fig. 8. For this and the following test problems, the function evaluation cost for each data point is computed as the total evaluation time (including both function and gradient evaluations) divided by the computational time of a single function evaluation, represented as function evaluation cost (num. of function evaluations).
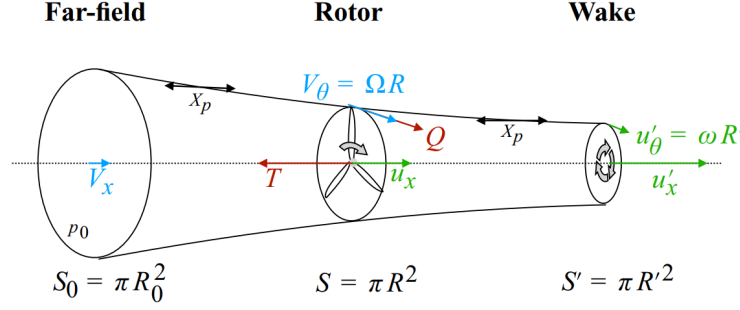

Fig. 7   Streamtube for the blade element momentum model [35]

| Uncertain inputs	Distributions
Rotor speed (RPM)	$U(1600, 1800)$
Axial free-stream velocity (m/s)	$U(50, 60)$
Propeller radius (m)	$U(0.79, 0.81)$
Flight altitude (m)	$U(9000, 10000)$

Table 1   Input parameters and ranges for the rotor analysis problem

From the results, it is evident that at a low evaluation cost (10s of equivalent model evaluations), UDR, GUDR, and NIPC are capable of delivering more accurate results compared to the Monte Carlo method in terms of both mean and standard deviation for this particular problem. In terms of estimating the mean of the output, UDR and GUDR yield identical UQ results when employing the same number of quadrature points in each dimension, but GUDR is computationally more demanding than UDR. This observation is aligned with our theoretical analysis, which affirms that both GUDR and UDR are fourth-order accurate and produce identical results when estimating the mean of the output. Additionally, both UDR and GUDR are more accurate than NIPC at low evaluation costs (10s of equivalent model evaluations).

As for the standard deviation of the output, UDR's accuracy is comparable to NIPC, while GUDR enhances the accuracy of UDR by almost an order of magnitude at the expense of a slightly increased computational cost. This phenomenon can be attributed to the fact that the GUDR approximation function results in fewer relative errors than the UDR approximation function when estimating the second and higher-order statistical moments. In this problem, with 20 equivalent model evaluations, GUDR can achieve a relative error of less than 0.1% for both mean and standard deviation of the output, making the GUDR method the most affordable UQ method to achieve this level of performance.

D. Aircraft design multidisciplinary model

fifth test problem is a 7D UQ problem derived from a practical aircraft design scenario. This problem involves a high-altitude, laser beam-powered unmanned aerial vehicle (UAV) cruising at a high altitude while receiving charge from a laser beam emitted by a ground station. The concept of the UAV is illustrated in Fig. 9, while the laser beam engagement scenario is depicted in Fig. 10. The computational model incorporates multiple disciplines, such as aerodynamics, structures, thermals, power beaming, performance, and weight models. This model calculates the required aircraft mass to meet certain criteria, including endurance, maximum stress, and static margin. The meshes for

(a) **Mean of the output** (b) **Standard deviation of the output**
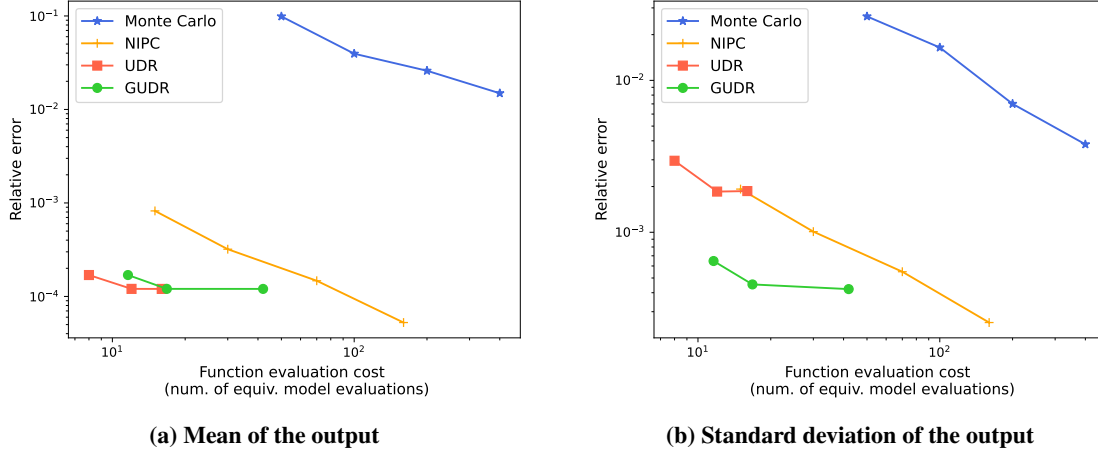
Fig. 8   Convergence plots for the rotor analysis problem

the aerodynamics and structure solvers are shown in Fig. 11. For further details on the computational model, we refer the reader to [38].

The objective of the UQ problem is to determine the mean and standard deviation of the required aircraft mass, taking into account seven uncertain inputs with uniform distributions. The specific uncertain inputs and their distributions are listed in Tab. 2. We use the UQ results generated from the Monte Carlo method with $10,000$ sample points as the reference results and depict the convergence plots of the four methods in Fig. 12.
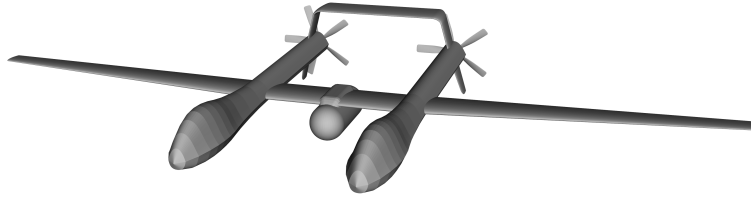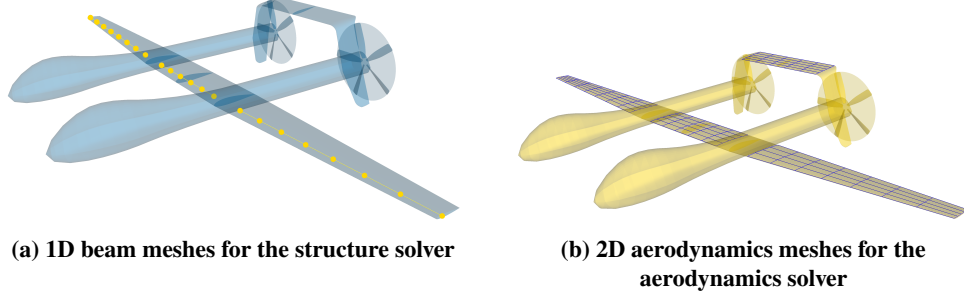

Fig. 9   The high-altitude, laser-powered airplane concept
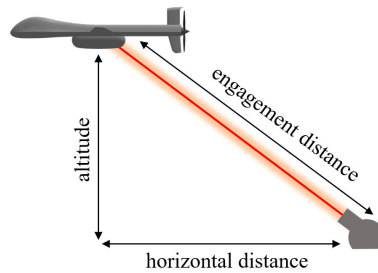
Fig. 10   The laser beam engagement scenario

Similar to the last test problem, the Monte Carlo method is outperformed by UDR, GUDR, and NIPC at low evaluation costs. When estimating the mean of the output, GUDR is slightly more expensive than UDR and both of them are more accurate than NIPC. However, when estimating the standard deviation of the output, UDR has comparable performance to NIPC at low function evaluation costs. In contrast, GUDR enhances the result of UDR by almost an order of magnitude with a slight increase in computational cost. These observations are also consistent with our theoretical results as GUDR can be significantly more accurate than UDR in estimating second and higher-order statistical moments of the output. As a result, in this problem, with the computational cost of 50 equivalent model evaluations, GUDR

13

(a) 1D beam meshes for the structure solver

(b) 2D aerodynamics meshes for the aerodynamics solver

Fig. 11   Meshes for structure and aerodynamics solvers

| Uncertain inputs	Distributions
Cruise mach number	$U(0.1, 0.2)$
Altitude(km)	$U(17, 19)$
Horizontal distance (km)	$U(250, 300)$
Pitch angle (deg)	$U(-2, 2)$
Payload mass (kg)	$U(50, 100)$
Aperture diameter (m)	$U(0.7, 0.8)$
Rotor speed (RPM)	$U(1200, 1500)$

Table 2   Input parameters and ranges for the aircraft design problem

can achieve a relative error of less than 1% for both the mean and standard deviation of the output, making the GUDR method the most affordable UQ method for achieving this level of performance.

V. Conclusion

In this paper, we introduced a UQ method, *gradient-enhanced univariate dimension reduction* (GUDR), for solving high-dimensional UQ problems. This method builds on the idea of univariate dimension reduction (UDR) and adds univariate gradient functions into the UDR approximation function so that it enhances its accuracy in approximating the original function. GUDR employs a computational graph transformation strategy to efficiently evaluate its approximation function on tensor-grid quadrature points to evaluate any statistical moment of the output. By using an automatic differentiation method to evaluate the gradient information, GUDR preserves the linear scaling cost of the UDR method.

The GUDR method was tested on five UQ problems: two mathematical functions in 2D and 3D, one mathematical

(a) Mean of the output

(b) Standard deviation of the output
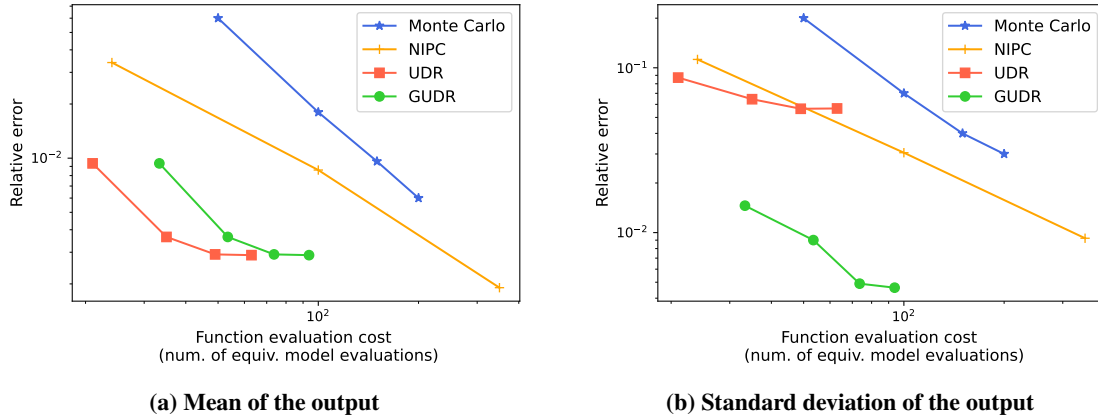
Fig. 12   Convergence plots for the aircraft design problem

14

function with varying dimensions, one 4D rotor analysis problem, and one 7D aircraft design problem. For the first two mathematical functions, GUDR surpassed UDR in accuracy for standard deviation estimation and matched the performance of the method of moments using the 3rd-order Taylor expansion. For the third mathematical function, GUDR exhibits linear scalability with the problem dimension in terms of function evaluation cost, which is less than four times that of UDR. In the 4D rotor analysis and 7D aircraft design problems, GUDR improved upon UDR's precision tenfold when estimating the output's standard deviation, achieving the most cost-effective performance among the UQ methods implemented.

A critical limitation of this work is that GUDR requires the evaluations of both first- and second-order gradients of the computational model using an automatic differentiation method, which is not a universal feature in all software environments. One important direction for future work is to evaluate the GUDR's performance in estimating third and higher-order statistical moments, such as skewness and kurtosis. Another promising area is to assess the GUDR's accuracy in estimating reliability metrics of the output by integrating it with polynomial chaos and kriging methods. Additionally, developing more accurate linear-scaling UQ methods based on the GUDR concept presents an interesting research opportunity. Furthermore, another important research question is how to develop a method that improves the accuracy of UDR in estimating the mean of the output by incorporating gradient information.

Appendix

A. Univariate dimension reduction in 2D case

In this section, we derive the residual errors of the univariate dimension reduction (UDR) method in estimating the mean and standard deviation for a 2D UQ problem. Consider an UQ problem that involves a bivariate function $f(u_1, u_2)$, the Taylor series expansion of f at $(u_1 = \mu_1, u_2 = \mu_2)$ can be expressed by

$$
\begin{aligned}
f(u_1, u_2) =& f(\mu_1, \mu_2) + \frac{\partial f}{\partial u_1}(\mu_1, \mu_2)(u_1 - \mu_1) + \frac{\partial f}{\partial u_2}(\mu_1, \mu_2)(u_2 - \mu_2) + \frac{1}{2!}\frac{\partial^2 f}{\partial u_1^2}(\mu_1, \mu_2)(u_1 - \mu_1)^2 \\
&+ \frac{1}{2!}\frac{\partial^2 f}{\partial u_2^2}(\mu_1, \mu_2)(u_2 - \mu_2)^2 + \frac{\partial^2 f}{\partial u_1 \partial u_2}(\mu_1, \mu_2)(u_1 - \mu_1)(u_2 - \mu_2) + \frac{1}{3!}\frac{\partial^3 f}{\partial u_1^3}(\mu_1, \mu_2)(u_1 - \mu_1)^3 \\
&+ \frac{1}{3!}\frac{\partial^3 f}{\partial u_2^3}(\mu_1, \mu_2)(u_2 - \mu_2)^3 + \frac{1}{2!}\frac{\partial^3 f}{\partial u_1^2 \partial u_2}(\mu_1, \mu_2)(u_1 - \mu_1)^2(u_2 - \mu_2) \\
&+ \frac{1}{2!}\frac{\partial^3 f}{\partial u_1 \partial u_2^2}(\mu_1, \mu_2)(u_1 - \mu_1)(u_2 - \mu_2)^2 + \frac{1}{4!}\frac{\partial^4 f}{\partial u_1^4}(\mu_1, \mu_2)(u_1 - \mu_1)^4 + \frac{1}{4!}\frac{\partial^4 f}{\partial u_2^4}(\mu_1, \mu_2)(u_2 - \mu_2)^4 \\
&+ \frac{1}{3!}\frac{\partial^4 f}{\partial u_1^3 \partial u_2}(\mu_1, \mu_2)(u_1 - \mu_1)^3(u_2 - \mu_2) + \frac{1}{3!}\frac{\partial^4 f}{\partial u_1 \partial u_2^3}(\mu_1, \mu_2)(u_1 - \mu_1)(u_2 - \mu_2)^3 \\
&+ \frac{1}{2!2!}\frac{\partial^4 f}{\partial u_1^2 \partial u_2^2}(\mu_1, \mu_2)(u_1 - \mu_1)^2(u_2 - \mu_2)^2 + \dots
\end{aligned}
\tag{26}
$$

In this case, the UDR method approximates the original function as

$$
\hat{f}(u_1, u_2) = f(u_1, \mu_2) + f(\mu_1, u_2) - f(\mu_1, \mu_2).
\tag{27}
$$

The Taylor series expansion of the UDR approximation, $\hat{f}$, at $(u_1 = \mu_1, u_2 = \mu_2)$ can be expressed as

$$
\begin{aligned}
\hat{f}(u_1, u_2) =& f(\mu_1, \mu_2) + \frac{\partial f}{\partial u_1}(\mu_1, \mu_2)(u_1 - \mu_1) + \frac{\partial f}{\partial u_2}(\mu_1, \mu_2)(u_2 - \mu_2) + \frac{1}{2!}\frac{\partial^2 f}{\partial u_1^2}(\mu_1, \mu_2)(u_1 - \mu_1)^2 \\
&+ \frac{1}{2!}\frac{\partial^2 f}{\partial u_2^2}(\mu_1, \mu_2)(u_2 - \mu_2)^2 + \frac{1}{3!}\frac{\partial^3 f}{\partial u_1^3}(\mu_1, \mu_2)(u_1 - \mu_1)^3 + \frac{1}{3!}\frac{\partial^3 f}{\partial u_2^3}(\mu_1, \mu_2)(u_2 - \mu_2)^3 \\
&+ \frac{1}{4!}\frac{\partial^4 f}{\partial u_1^4}(\mu_1, \mu_2)(u_1 - \mu_1)^4 + \frac{1}{4!}\frac{\partial^4 f}{\partial u_2^4}(\mu_1, \mu_2)(u_2 - \mu_2)^4 + \dots
\end{aligned}
\tag{28}
$$

15

If we compare the Taylor series expansion of the original function with the UDR approximation, all of the Taylor series terms in (28) are contained in (26), and the residual errors can be expressed as

$$
f\left(u_1, u_2\right) - \hat{f}\left(u_1, u_2\right) = \frac{\partial^2 f}{\partial u_1 \partial u_2}(\mu_1, \mu_2)(u_1 - \mu_1)(u_2 - \mu_2) + \frac{1}{2!}\frac{\partial^3 f}{\partial u_1^2 \partial u_2}(\mu_1, \mu_2)(u_1 - \mu_1)^2(u_2 - \mu_2)
$$

$$
+ \frac{1}{2!}\frac{\partial^3 f}{\partial u_1 \partial u_2^2}(\mu_1, \mu_2)(u_1 - \mu_1)(u_2 - \mu_2)^2 + \frac{1}{3!}\frac{\partial^4 f}{\partial u_1^3 \partial u_2}(\mu_1, \mu_2)(u_1 - \mu_1)^3(u_2 - \mu_2)
$$

$$
+ \frac{1}{2!2!}\frac{\partial^4 f}{\partial u_1^2 \partial u_2^2}(\mu_1, \mu_2)(u_1 - \mu_1)^2(u_2 - \mu_2)^2 + \frac{1}{3!}\frac{\partial^4 f}{\partial u_1 \partial u_2^3}(\mu_1, \mu_2)(u_1 - \mu_1)(u_2 - \mu_2)^3 + \ldots
$$

$$(29)$$

From (29), we observe that the UDR approximation is only second-order accurate compared to the original function. However, when it comes to estimating the mean of the output, the residual errors can be expressed as

$$
\mathbb{E}[f(U_1, U_2)] - \mathbb{E}[\hat{f}(U_1, U_2)] = \int_{\Gamma_1}\int_{\Gamma_2}\left(f(u_1, u_2) - \hat{f}(u_1, u_2)\right)\rho(u_1)\rho(u_2)du_1du_2
$$

$$
= \frac{1}{2!2!}\frac{\partial^4 f}{\partial u_1^2 \partial u_2^2}(\mu_1, \mu_2)\mathbb{E}[(u_1 - \mu_1)^2(u_2 - \mu_2)^2] + \ldots
$$

$$(30)$$

In comparison, for a 3rd-order Taylor series method which approximates the function as

$$
\tilde{f}\left(u_1, u_2\right) = f(\mu_1, \mu_2) + \frac{\partial f}{\partial u_1}(\mu_1, \mu_2)(u_1 - \mu_1) + \frac{\partial f}{\partial u_2}(\mu_1, \mu_2)(u_2 - \mu_2) + \frac{1}{2!}\frac{\partial^2 f}{\partial u_1^2}(\mu_1, \mu_2)(u_1 - \mu_1)^2
$$

$$
+ \frac{1}{2!}\frac{\partial^2 f}{\partial u_2^2}(\mu_1, \mu_2)(u_2 - \mu_2)^2 + \frac{\partial^2 f}{\partial u_1 \partial u_2}(\mu_1, \mu_2)(u_1 - \mu_1)(u_2 - \mu_2) + \frac{1}{3!}\frac{\partial^3 f}{\partial u_1^3}(\mu_1, \mu_2)(u_1 - \mu_1)^3
$$

$$
+ \frac{1}{3!}\frac{\partial^3 f}{\partial u_2^3}(\mu_1, \mu_2)(u_2 - \mu_2)^3 + \frac{1}{2!}\frac{\partial^3 f}{\partial u_1^2 \partial u_2}(\mu_1, \mu_2)(u_1 - \mu_1)^2(u_2 - \mu_2)
$$

$$
+ \frac{1}{2!}\frac{\partial^3 f}{\partial u_1 \partial u_2^2}(\mu_1, \mu_2)(u_1 - \mu_1)(u_2 - \mu_2)^2
$$

$$(31)$$

The residual errors of the 3rd-order Taylor series method when estimating the mean of the output can be expressed as

$$
\mathbb{E}[f(U_1, U_2)] - \mathbb{E}[\tilde{f}(U_1, U_2)] = \frac{1}{2!2!}\frac{\partial^4 f}{\partial u_1^2 \partial u_2^2}(\mu_1, \mu_2)\mathbb{E}[(u_1 - \mu_1)^2(u_2 - \mu_2)^2] + \frac{1}{4!}\frac{\partial^4 f}{\partial u_1^4}(\mu_1, \mu_2)\mathbb{E}[(u_1 - \mu_1)^4]
$$

$$
+ \frac{1}{4!}\frac{\partial^4 f}{\partial u_2^4}(\mu_1, \mu_2)\mathbb{E}[(u_2 - \mu_2)^4] + \ldots
$$

$$(32)$$

Comparing the residual errors of the 3rd-order Taylor series method in (32) with the residual errors of the UDR in (30). Both of the methods are fourth-order accurate in estimating the mean of the output, but the relative errors of the UDR comprise fewer fourth-order integration terms compared with 3rd-order Taylor series expansion. When it comes to estimating the second-order statistical moments, the relative errors of the UDR can be expressed as:

$$
\mathbb{E}[f(U_1, U_2)^2] - \mathbb{E}[\hat{f}(U_1, U_2)^2] = \int_{\Gamma_1}\int_{\Gamma_2}\left(f(u_1, u_2)^2 - \hat{f}(u_1, u_2)^2\right)\rho(u_1)\rho(u_2)du_1du_2
$$

$$
= \left(\frac{\partial^2 f}{\partial u_1 \partial u_2}(\mu_1, \mu_2)\right)^2 \mathbb{E}[(u_1 - \mu_1)^2(u_2 - \mu_2)^2]
$$

$$
+ \left(\frac{\partial y}{\partial u_2}(\mu_1, \mu_2)\frac{\partial^3 y}{\partial u_1^2 \partial u_2}(\mu_1, \mu_2) + \frac{\partial y}{\partial u_1}(\mu_1, \mu_2)\frac{\partial^3 y}{\partial u_1 \partial u_2^2}(\mu_1, \mu_2)\right)\mathbb{E}[(u_1 - \mu_1)^2(u_2 - \mu_2)^2]
$$

$$
+ \frac{1}{2!}f(\mu_1, \mu_2)\frac{\partial^4 f}{\partial u_1^2 \partial u_2^2}(\mu_1, \mu_2)\mathbb{E}[(u_1 - \mu_1)^2(u_2 - \mu_2)^2] + \ldots
$$

$$(33)$$

The residual errors for 3rd order Taylor series method can be expressed as:

$$\mathbb{E}[f(U_1, U_2)^2] - \mathbb{E}[\tilde{f}(U_1, U_2)^2] = \frac{1}{2!} f(\mu_1, \mu_2) \frac{\partial^4 f}{\partial u_1^2 \partial u_2^2}(\mu_1, \mu_2)\mathbb{E}[(u_1 - \mu_1)^2(u_2 - \mu_2)^2] + \dots \quad (34)$$

Comparing the residual errors of the 3rd-order Taylor series expansion method in (41) with the residual errors of the UDR in (33). Both of the methods are also fourth-order accurate in estimating the second-order moment of the output, but the relative errors of the UDR comprise more fourth-order integration terms compared with 3rd-order Taylor series expansion.

B. Gradient-enhanced univariate dimension reduction in 2D case

In this section, we derive the residual errors of the gradient-enhanced univariate dimension reduction (GUDR) method in estimating the mean and standard deviation for a 2D UQ problem. Consider an UQ problem that involves a bivariate function $f(u_1, u_2)$, the GUDR method approximates the original function as

$$\bar{f}(u_1, u_2) = y(u_1, \mu_2) + y(\mu_1, \mu_2) - y(\mu_1, \mu_2) + (u_1 - \mu_1)\left(\frac{\partial f}{\partial u_1}(\mu_1, u_2) - \frac{\partial f}{\partial u_1}(\mu_1, \mu_2)\right)$$
$$+ (u_2 - \mu_2)\left(\frac{\partial f}{\partial u_2}(u_1, \mu_2) - \frac{\partial f}{\partial u_2}(\mu_1, \mu_2)\right) - \frac{\partial^2 f}{\partial u_1 u_2}(\mu_1, \mu_2)(u_1 - \mu_1)(u_2 - \mu_2). \quad (35)$$

The Taylor series of the involved univariate gradient terms at $(u_1 = \mu_1, u_2 = \mu_2)$ can be expressed by

$$\frac{\partial f}{\partial u_1}(\mu_1, u_2) = \frac{\partial f}{\partial u_1}(\mu_1, \mu_2) + \frac{\partial^2 f}{\partial u_1 u_2}(\mu_1, \mu_2)(u_2 - \mu_2) + \frac{1}{2!}\frac{\partial^3 f}{\partial u_1 u_2^2}(\mu_1, \mu_2)(u_2 - \mu_2)^2 + \frac{1}{3!}\frac{\partial^4 f}{\partial u_1 u_2^3}(\mu_1, \mu_2)(u_2 - \mu_2)^3 + \dots, \quad (36)$$

$$\frac{\partial f}{\partial u_2}(u_1, \mu_2) = \frac{\partial f}{\partial u_2}(\mu_1, \mu_2) + \frac{\partial^2 f}{\partial u_1 u_2}(\mu_1, \mu_2)(u_1 - \mu_1) + \frac{1}{2!}\frac{\partial^3 f}{\partial u_1^2 u_2}(\mu_1, \mu_2)(u_1 - \mu_1)^2 + \frac{1}{3!}\frac{\partial^4 f}{\partial u_1^3 u_2}(\mu_1, \mu_2)(u_1 - \mu_1)^3 + \dots. \quad (37)$$

The Taylor series of GUDR approximation function, $\bar{f}$ can be expressed as

$$\bar{f}(u_1, u_2) = f(\mu_1, \mu_2) + \frac{\partial f}{\partial u_1}(\mu_1, \mu_2)(u_1 - \mu_1) + \frac{\partial f}{\partial u_2}(\mu_1, \mu_2)(u_2 - \mu_2) + \frac{1}{2!}\frac{\partial^2 f}{\partial u_1^2}(\mu_1, \mu_2)(u_1 - \mu_1)^2$$

$$+ \frac{1}{2!}\frac{\partial^2 f}{\partial u_2^2}(\mu_1, \mu_2)(u_2 - \mu_2)^2 + \frac{\partial^2 f}{\partial u_1 \partial u_2}(\mu_1, \mu_2)(u_1 - \mu_1)(u_2 - \mu_2) + \frac{1}{3!}\frac{\partial^3 f}{\partial u_1^3}(\mu_1, \mu_2)(u_1 - \mu_1)^3$$

$$+ \frac{1}{3!}\frac{\partial^3 f}{\partial u_2^3}(\mu_1, \mu_2)(u_2 - \mu_2)^3 + \frac{1}{2!}\frac{\partial^3 f}{\partial u_1^2 \partial u_2}(\mu_1, \mu_2)(u_1 - \mu_1)^2(u_2 - \mu_2)$$

$$+ \frac{1}{2!}\frac{\partial^3 f}{\partial u_1 \partial u_2^2}(\mu_1, \mu_2)(u_1 - \mu_1)(u_2 - \mu_2)^2 + \frac{1}{4!}\frac{\partial^4 f}{\partial u_1^4}(\mu_1, \mu_2)(u_1 - \mu_1)^4 + \frac{1}{4!}\frac{\partial^4 f}{\partial u_2^4}(\mu_1, \mu_2)(u_2 - \mu_2)^4$$

$$+ \frac{1}{3!}\frac{\partial^4 f}{\partial u_1^3 \partial u_2}(\mu_1, \mu_2)(u_1 - \mu_1)^3(u_2 - \mu_2) + \frac{1}{3!}\frac{\partial^4 f}{\partial u_1 \partial u_2^3}(\mu_1, \mu_2)(u_1 - \mu_1)(u_2 - \mu_2)^3 + \dots \quad (38)$$

If we compare the Taylor series expansion of the original function with the GUDR approximation, all of the Taylor series terms in (38) are contained in (26), and the residual errors can be expressed as

$$f(u_1, u_2) - \bar{f}(u_1, u_2) = \frac{1}{2!2!}\frac{\partial^4 f}{\partial u_1^2 \partial u_2^2}(0, 0)u_1^2 u_2^2 + \dots \quad (39)$$

which only include fourth and higher-order terms. When we estimate the first-order statistical moment of the output, the GUDR approximation generates the same results as the UDR as

$$\mathbb{E}[f(u_1, u_2)] - \mathbb{E}[\bar{f}(u_1, u_2)] = \frac{1}{2!2!}\frac{\partial^4 f}{\partial u_1^2 \partial u_2^2}(\mu_1, \mu_2)\mathbb{E}[(u_1 - \mu_1)^2(u_2 - \mu_2)^2] + \dots \quad (40)$$

17

When we estimate the second-order statistical moment of the output, the residual errors for GUDR approximation can be expressed as

$$\mathbb{E}[f(u_1, u_2)^2] - \mathbb{E}[\bar{f}(u_1, u_2)^2] = \frac{1}{2!} f(\mu_1, \mu_2) \frac{\partial^4 f}{\partial u_1^2 \partial u_2^2}(\mu_1, \mu_2) \mathbb{E}[(u_1 - \mu_1)^2 (u_2 - \mu_2)^2] + \ldots. \tag{41}$$

Compared to the residual errors of the 3rd-order Taylor series method, shown in (41), the residual errors of the GUDR approximation contain the same fourth-order integration terms.

C. Example code to implement computational graph transformation in GUDR

In this section, we provide the example code to implement the computational graph transformation method in GUDR for a 2D UQ problem. The Python code below assumes we have the univariate model and gradient functions' evaluations on quadrature points of each input dimension, along with the function and gradient evaluations on the mean of the input. The code shows how to use NumPy's Einstein summation function to expand the sizes of certain inputs so that we can generate the correct tensor-grid evaluations for the GUDR approximation function.

```python
import numpy as np
k # number of quadrature points in each dimension
u_1 # quadrature points in u_1, size: (k,1)
u_2  # quadrature points in u_2, size:  (k,1)
f1_u1 #  quadrature points evaluations of f_1(u_1), size (k,1)
f2_u2 #  quadrature points evaluations of f_2(u_2), size (k,1)
df1_du # quadrature points evaluations of df_1/du(u_1), size (k,2)
df2_du # quadrature points evaluations of df_2/du(u_2), size (k,2)
mean_u # mean of the inputs, size(2,)
f_mean_u = # function evaluation on mean of the inputs, size (1,)
df_du_mean_u = # gradient evaluation on mean of the inputs, size (2,)
d2f_du2_mean_u = # Hessian evaluation on mean of the inputs, size (2,2)
total_points = k**2 # total input points k**2

# Use Einsum and reshape operations to expand the size of certain inputs
u_1 = np.einsum('i...,p...->ip...', u_1 , np.ones(k))
u_1 = np.reshape(u_1 , (total_points,))
u_2 = np.einsum('p...,i...->ip...', u_2 , np.ones(k))
u_2 = np.reshape(u_2 , (total_points,))
f1_u1 = np.einsum('i...,p...->ip...', f1_u1, np.ones(k))
f1_u1 = np.reshape(f1_u1, (total_points,))
f2_u2 = np.einsum('p...,i...->ip...', f2_u2, np.ones(k))
f2_u2 = np.reshape(f2_u2, (total_points,))
df1_du = np.einsum('i...,p...->ip...', df1_du, np.ones(k))
df1_du = np.reshape(df1_du, (total_points, 2))
df2_du = np.einsum('p...,i...->ip...', df2_du, np.ones(k))
df2_du = np.reshape(df2_du, (total_points, 2))

# Evaluate GUDR approximation function on tensor-grid quadrature points
f_gudr = np.zeros(total_points,)
for i in range(total_points):
  u =  np.array([u_1[i], u_2[i]])
  a1 = np.array([0, u_2[i]- mean_u[1]])
  a2 = np.array([u_1[i]- mean_u[0], 0])
  term1 = f1_u1[i] + f2_u2[i] - f_mean_u
  term2 = np.dot(a1, df1_du[i,:]) + np.dot(a2, df2_du[i,:])- np.dot((u-mean_u), df_du_mean_u)
  term3 = -0.5*np.einsum('...i,...i->...', (u-mean_u).T.dot(d2f_du2_mean_u - np.diag(np.diag(
                                           d2f_du2_mean_u))), (u-mean_u))
  f_gudr[i] = term1 + term2+ term3

# f_gudr: GUDR approximation function evaluations on tensor-grid quadrature points, size (k^2,)
```

Acknowledgments

References

[1] Joslyn, S., and Savelli, S., "Communicating forecast uncertainty: Public perception of weather forecast uncertainty," *Meteorological Applications*, Vol. 17, No. 2, 2010, pp. 180–195. https://doi.org/10.1002/met.190.

[2] Pappenberger, F., Beven, K. J., Hunter, N. M., Bates, P. D., Gouweleeuw, B. T., Thielen, J., and de Roo, A. P. J., "Cascading model uncertainty from medium range weather forecasts (10 days) through a rainfall-runoff model to flood inundation predictions within the European Flood Forecasting System (EFFS)," *Hydrology and Earth System Sciences*, Vol. 9, No. 4, 2005, pp. 381–393. https://doi.org/10.5194/hess-9-381-2005.

[3] Hüllermeier, E., and Waegeman, W., "Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods," *Machine Learning*, Vol. 110, 2021, pp. 457–506. https://doi.org/10.1007/s10994-021-05946-3.

[4] Wan, H.-P., Mao, Z., Todd, M. D., and Ren, W.-X., "Analytical uncertainty quantification for modal frequencies with structural parameter uncertainty using a Gaussian process metamodel," *Engineering Structures*, Vol. 75, 2014, pp. 577–589. https://doi.org/10.1016/j.engstruct.2014.06.028.

[5] Hu, Z., Mahadevan, S., and Ao, D., "Uncertainty aggregation and reduction in structure–material performance prediction," *Computational Mechanics*, Vol. 61, No. 1, 2018, pp. 237–257. https://doi.org/10.1007/s00466-017-1448-6.

[6] Ng, L. W., and Willcox, K. E., "Monte Carlo information-reuse approach to aircraft conceptual design optimization under uncertainty," *Journal of Aircraft*, Vol. 53, No. 2, 2016, pp. 427–438. https://doi.org/10.2514/1.C033352.

[7] Wang, B., Orndorff, N. C., Joshy, A. J., and Hwang, J. T., "Graph-accelerated large-scale multidisciplinary design optimization under uncertainty of a laser-beam-powered aircraft," *AIAA SCITECH 2024 Forum*, 2024, p. 0169. https://doi.org/10.2514/6.2024-0169.

[8] Lim, D., Kim, H., and Yee, K., "Uncertainty propagation in flight performance of multirotor with parametric and model uncertainties," *Aerospace Science and Technology*, Vol. 122, 2022, p. 107398. https://doi.org/10.1016/j.ast.2022.107398.

[9] Wooldridge, J. M., "Applications of generalized method of moments estimation," *Journal of Economic perspectives*, Vol. 15, No. 4, 2001, pp. 87–100. https://doi.org/10.1257/jep.15.4.87.

[10] Fragkos, K., Papoutsis-Kiachagias, E., and Giannakoglou, K., "pFOSM: An efficient algorithm for aerodynamic robust design based on continuous adjoint and matrix-vector products," *Computers & Fluids*, Vol. 181, 2019, pp. 57–66. https://doi.org/10.1016/j.compfluid.2019.01.016.

[11] Peherstorfer, B., Willcox, K., and Gunzburger, M., "Optimal model management for multifidelity Monte Carlo estimation," *SIAM Journal on Scientific Computing*, Vol. 38, No. 5, 2016, pp. A3163–A3194. https://doi.org/10.1137/15M1046472.

[12] Peherstorfer, B., Willcox, K., and Gunzburger, M., "Survey of multifidelity methods in uncertainty propagation, inference, and optimization," *Siam Review*, Vol. 60, No. 3, 2018, pp. 550–591. https://doi.org/10.1137/16M1082469.

[13] Tabandeh, A., Jia, G., and Gardoni, P., "A review and assessment of importance sampling methods for reliability analysis," *Structural Safety*, Vol. 97, 2022, p. 102216. https://doi.org/10.1016/j.strusafe.2022.102216.

[14] Kaymaz, I., "Application of kriging method to structural reliability problems," *Structural safety*, Vol. 27, No. 2, 2005, pp. 133–151. https://doi.org/10.1016/j.strusafe.2004.09.001.

[15] Hu, Z., and Mahadevan, S., "A single-loop kriging surrogate modeling for time-dependent reliability analysis," *Journal of Mechanical Design*, Vol. 138, No. 6, 2016, p. 061406. https://doi.org/10.1115/1.4033428.

[16] Rumpfkeil, M. P., "Optimizations under uncertainty using gradients, Hessians, and surrogate models," *AIAA journal*, Vol. 51, No. 2, 2013, pp. 444–451. https://doi.org/10.2514/1.J051847.

[17] Hosder, S., Walters, R., and Perez, R., "A non-intrusive polynomial chaos method for uncertainty propagation in CFD simulations," *44th AIAA aerospace sciences meeting and exhibit*, 2006, p. 891. https://doi.org/10.2514/6.2006-891.

[18] Jones, B. A., Doostan, A., and Born, G. H., "Nonlinear propagation of orbit uncertainty using non-intrusive polynomial chaos," *Journal of Guidance, Control, and Dynamics*, Vol. 36, No. 2, 2013, pp. 430–444. https://doi.org/10.2514/1.57599.

[19] Keshavarzzadeh, V., Fernandez, F., and Tortorelli, D. A., "Topology optimization under uncertainty via non-intrusive polynomial chaos expansion," *Computer Methods in Applied Mechanics and Engineering*, Vol. 318, 2017, pp. 120–147. https://doi.org/10.1016/j.cma.2017.01.019.

[20] Morio, J., "Global and local sensitivity analysis methods for a physical system," *European journal of physics*, Vol. 32, No. 6, 2011, p. 1577. https://doi.org/10.1088/0143-0807/32/6/011.

[21] Chun, H., and Keleş, S., "Sparse partial least squares regression for simultaneous dimension reduction and variable selection," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, Vol. 72, No. 1, 2010, pp. 3–25. https://doi.org/10.1111/j.1467-9868.2009.00723.x.

[22] Constantine, P. G., Dow, E., and Wang, Q., "Active subspace methods in theory and practice: applications to kriging surfaces," *SIAM Journal on Scientific Computing*, Vol. 36, No. 4, 2014, pp. A1500–A1524. https://doi.org/10.1137/130916138.

[23] Rahman, S., and Xu, H., "A univariate dimension-reduction method for multi-dimensional integration in stochastic mechanics," *Probabilistic Engineering Mechanics*, Vol. 19, No. 4, 2004, pp. 393–408. https://doi.org/10.1016/j.probengmech.2004.04.003.

[24] Piric, K., "Reliability analysis method based on determination of the performance functionâĂŹs PDF using the univariate dimension reduction method," *Structural safety*, Vol. 57, 2015, pp. 18–25. https://doi.org/10.1016/j.strusafe.2015.07.005.

[25] Huang, B., and Du, X., "Uncertainty analysis by dimension reduction integration and saddlepoint approximations," 2006. https://doi.org/10.1115/1.2118667.

[26] Lee, I., Choi, K., Du, L., and Gorsich, D., "Dimension reduction method for reliability-based robust design optimization," *Computers & Structures*, Vol. 86, No. 13-14, 2008, pp. 1550–1562. https://doi.org/10.1016/j.compstruc.2007.05.020.

[27] Xu, H., and Rahman, S., "A generalized dimension-reduction method for multidimensional integration in stochastic mechanics," *International Journal for Numerical Methods in Engineering*, Vol. 61, No. 12, 2004, pp. 1992–2019. https://doi.org/10.1002/nme.1135.

[28] Wang, B., Sperry, M., Gandarillas, V. E., and Hwang, J. T., "Accelerating model evaluations in uncertainty propagation on tensor grids using computational graph transformations," *Aerospace Science and Technology*, Vol. 145, 2024, p. 108843. https://doi.org/10.1016/j.ast.2023.108843.

[29] Wang, B., Sperry, M., Gandarillas, V. E., and Hwang, J. T., "Efficient uncertainty propagation through computational graph modification and automatic code generation," *AIAA AVIATION 2022 Forum*, 2022, p. 3997. https://doi.org/10.2514/6.2022-3997.

[30] Wang, B., Orndorff, N. C., Sperry, M., and Hwang, J. T., "Extension of graph-accelerated non-intrusive polynomial chaos to high-dimensional uncertainty quantification through the active subspace method," *arXiv preprint arXiv:2405.05556*, 2024. https://doi.org/10.48550/arXiv.2405.05556.

[31] Wang, B., Orndorff, N. C., and Hwang, J. T., "Graph-accelerated non-intrusive polynomial chaos expansion using partially tensor-structured quadrature rules," *arXiv preprint arXiv:2403.15614*, 2024. https://doi.org/10.48550/arXiv.2403.15614.

[32] Wang, B., "Graph-accelerated uncertainty propagation for large-scale multidisciplinary design, analysis, and optimization under uncertainty," Ph.D. thesis, UC San Diego, 2024.

[33] Gandarillas, V., Joshy, A. J., Sperry, M. Z., Ivanov, A. K., and Hwang, J. T., "A graph-based methodology for constructing computational models that automates adjoint-based sensitivity analysis," *Structural and Multidisciplinary Optimization*, Vol. 67, No. 5, 2024, p. 76. https://doi.org/10.1007/s00158-024-03792-0.

[34] Christianson, B., "Automatic Hessians by reverse accumulation," *IMA Journal of Numerical Analysis*, Vol. 12, No. 2, 1992, pp. 135–150. https://doi.org/10.1093/imanum/12.2.135.

[35] Ruh, M. L., and Hwang, J. T., "Fast and Robust Computation of Optimal Rotor Designs Using Blade Element Momentum Theory," *AIAA Journal*, Vol. 61, No. 9, 2023, pp. 4096–4111. https://doi.org/10.2514/1.J062611.

[36] Sperry, M., Kondap, K., and Hwang, J. T., "Automatic adjoint sensitivity analysis of models for large-scale multidisciplinary design optimization," *AIAA AVIATION 2023 Forum*, 2023, p. 3721. https://doi.org/10.2514/6.2023-3721.

[37] Feinberg, J., and Langtangen, H. P., "Chaospy: An open source tool for designing methods of uncertainty quantification," *Journal of Computational Science*, Vol. 11, 2015, pp. 46–57. https://doi.org/10.1016/j.jocs.2015.08.008.

[38] Orndorff, N. C., Wang, B., Ruh, M. L., Fletcher, A., and Hwang, J. T., "Gradient-based sizing optimization of power-beaming-enabled aircraft," *AIAA AVIATION 2023 Forum*, 2023, p. 4019. https://doi.org/10.2514/6.2023-4019.