AIAA SciTech Forum
8-12 January 2024, Orlando, FL
AIAA SCITECH 2024 Forum

10.2514/6.2024-1234

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, 92093

The univariate dimension reduction (UDR) method stands as an economical solution for addressing the challenges of high-dimensional uncertainty quantification (UQ). 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 the tensor-grid input-output data can be easily used to compute any risk measure of the output. With an efficient automatic differentiation method to compute the gradients, our methods preserve UDR's linear cost-scaling with problem dimension. Numerical results show that the GUDR is more accurate than UDR in estimating the standard deviation of the output and has comparable performance with the method of moments using 3rd-order Taylor series expansion.

I. Introduction

Uncertainties are inherent in numerous scientific and engineering problems, from weather forecasting [1, 2] and machine learning [3] to structural analysis [4, 5] and aircraft design [6, 7]. They can have a considerable effect on a system's behavior or performance. Forward uncertainty quantification (UQ), or uncertainty propagation, is essential for evaluating 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, referred to as epistemic uncertainties. By quantifying the impact of this uncertainty, UQ provides critical support for decision-making and risk assessment. In this paper, we consider a UQ problem within a probabilistic framework. We aim to estimate the QoI's statistical moments or other risk measures like the full probability distribution or conditional value at risk.

The method 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 methods for directly solving UQ problems include the method of moments, polynomial chaos, kriging, and Monte Carlo. The method of moments [8, 9] uses Taylor series approximations to analytically compute the statistical moments, which is cost-effective but may lack accuracy for large input variances and becomes impractical at third or higher expansion orders due to increased computational demand. Monte Carlo methods constitute another widely used approach for solving high-dimensional UQ problems as its convergence rate is unaffected by the dimensionality of the input space. Recent advancements seek to enhance its efficiency through multi-fidelity [10, 11] and importance sampling techniques [12]. 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 [13, 14] or optimization amid uncertainty [15]. The polynomial chaos approach models the QoI using orthogonal polynomials based on the distributions of the inputs,

*Ph.D Candidate, Department of Mechanical and Aerospace Engineering, AIAA Student Member.
†Ph.D Student, Department of Mechanical and Aerospace Engineering, AIAA Student Member.
‡Ph.D Student, Department of Mechanical and Aerospace Engineering, AIAA Student Member.
§Assistant Professor, Department of Mechanical and Aerospace Engineering, AIAA Member.

leveraging the smoothness in the random space for swift convergence with sampling or integration [16–18]. 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 [19], partial least squares (PLS) [20], active subspace (AS) [21], and univariate dimension reduction (UDR) [22] 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. Conversely, UDR simplifies the function by approximating it as a sum of univariate functions, transforming a multidimensional integration problem into multiple one-dimensional problems that are collectively much cheaper to solve. Consequently, the computational load of UDR increases only linearly with the number of dimensions, presenting a very cost-effective method for high-dimensional UQ problems. UDR has also been adapted for calculating complete probability distributions [23] and reliability metrics of outputs [24, 25]. 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 [26]. Although these adjustments provide better accuracy, they compromise UDR's linear scalability—a trade-off that requires careful consideration in the method's application.

The authors recently introduced a novel method known as Accelerated Model evaluations on Tensor grids using Computational graph transformations (AMTC) [27, 28]. This method minimizes the model evaluation cost on tensor-grid inputs by modifying the computational graph of the model from the middle-end of a modeling language's three-stage compiler to eliminate the redundant evaluations on the operational level that are incurred by the tensor structure of the inputs. The existing work applies AMTC to integration-based NIPC, assuming a certain type of sparsity is present in the computational graph of the original function. In some problems, this sparsity enables speedups of multiple orders of magnitude [29, 30]. 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), with the aim 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. The theoretical results show that 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 risk measures of the output, we propose to use the AMTC strategy to effectively generate the evaluations of the GUDR approximation function on tensor-grid quadrature points. This tensor-grid input-output data can be easily used to compute any risk measure 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 three UQ problems; this includes one 2D and one 3D mathematical function, and one 7D problem derived from a practical aircraft design scenario. The results indicate that on the 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 7D aircraft design problem, GUDR enhances the accuracy of UDR by an order of magnitude in estimating the standard deviation and can achieve a relative error of less than 1% for both mean and standard deviation of the output with computational cost equivalent to 50 model evaluations.

This paper is organized as follows. Section II gives 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 uncertainties associated with the inputs are expressed as a stochastic vector denoted by $U := [U_1, \ldots, U_d]^T$, under the assumption that these random variables are mutually independent. The stochastic input vector adheres to the probability distribution $\rho(u)$ with support Γ. The UQ problem aims to

compute the risk measures of the output random variable, $f(U)$. Given that the mean values of the uncertain inputs are $\mu := [\mu_1, \ldots, \mu_d]^T$, the univariate dimension reduction (UDR) [22] method approximates the original function $f(u)$ as additive terms of univariate functions:

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

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

$$\begin{aligned} f(u) = f(\mu) &+ \sum_{j=1}^{\infty} \sum_{i=1}^{d} \frac{1}{j!} \frac{\partial^j f}{\partial u_i^j}(\mu)(u_i - \mu_i)^j + \sum_{j_2=1}^{\infty} \sum_{j_1=1}^{\infty} \frac{1}{j_1! j_2!} \sum_{i_1=1}^{d-1} \sum_{i_2>i_1}^{d} \frac{\partial^{j_1+j_2} f}{\partial u_{i_1}^{j_1} \partial u_{i_2}^{j_2}}(\mu)(u_{i_1} - \mu_{i_1})^{j_1}(u_{i_2} - \mu_{i_2})^{j_2} \\ &+ \sum_{j_3=1}^{\infty} \sum_{j_2=1}^{\infty} \sum_{j_1=1}^{\infty} \frac{1}{j_1! j_2! j_3!} \sum_{i_1=1}^{d-2} \sum_{i_2>i_1}^{d-1} \sum_{i_3>i_2}^{d} \frac{\partial^{j_1+j_2+j_3} f}{\partial u_{i_1}^{j_1} \partial u_{i_2}^{j_2} \partial u_{i_3}^{j_3}}(\mu)(u_{i_1} - \mu_{i_1})^{j_1}(u_{i_2} - \mu_{i_2})^{j_2}(u_{i_3} - \mu_{i_3})^{j_3} + \ldots \end{aligned} \tag{2}$$

The Taylor series expansion of the UDR approximation, $\hat{f}(u)$, 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 expansion of the original function with the UDR approximation, all of the Taylor series terms in (3) are contained in (2), and the residual errors are

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

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

$$\begin{aligned} \mathbb{E}[f(U)] - \mathbb{E}[\hat{f}(U)] &= \int_{\Gamma_1} \ldots \int_{\Gamma_d} \left(f(u) - \hat{f}(u) \right) \rho(u) du_1 \ldots du_d \\ &= \frac{1}{2!2!} \sum_{i_1=1}^{d-1} \sum_{i_2>i_1}^{d} \frac{\partial^4 f}{\partial u_{i_1}^2 \partial u_{i_2}^2}(\mu) \mathbb{E}[(u_{i_1} - \mu_{i_1})^2 (u_{i_2} - \mu_{i_2})^2] + \ldots, \end{aligned} \tag{5}$$

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

$$\begin{aligned} \tilde{f}(u) = f(\mu) &+ \sum_{j=1}^{3} \sum_{i=1}^{d} \frac{1}{j!} \frac{\partial^j f}{\partial u_i^j}(\mu)(u_i - \mu_i)^j + \sum_{j_1=1}^{2} \sum_{j_2=1}^{j_1+j_2 \le 3} \frac{1}{j_1! j_2!} \sum_{i_1=1}^{d-1} \sum_{i_2>i_1}^{d} \frac{\partial^{j_1+j_2} f}{\partial u_{i_1}^{j_1} \partial u_{i_2}^{j_2}}(\mu)(u_{i_1} - \mu_{i_1})^{j_1}(u_{i_2} - \mu_{i_2})^{j_2} \\ &+ \sum_{i_1=1}^{d-2} \sum_{i_2>i_1}^{d-1} \sum_{i_3>i_2}^{d} \frac{\partial^3 f}{\partial u_{i_1} \partial u_{i_2} \partial u_{i_3}}(\mu)(u_{i_1} - \mu_1)(u_{i_2} - \mu_2)(u_{i_3} - \mu_3). \end{aligned} \tag{6}$$

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

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

Upon comparing the UDR's residual errors in (5) with those of the 3rd-order Taylor expansion 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 yields a more accurate result than the 3rd-order Taylor expansion when estimating the mean of the output.

Nevertheless, for higher-order statistical moments or other risk measures, 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:

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

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

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

Comparing the residual errors of the 3rd-order Taylor series expansion 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 computing 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 the 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 replaced with 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, which is much more expensive. 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 [22] without requiring extra model evaluations.

B. Computational graph transformation to accelerate tensor-grid evaluations

The authors recently proposed the AMTC (Accelerated Model evaluations on Tensor grids using Computational graph transformations) method in [27]. AMTC is a graph transformation method to minimize the computational 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 tackles 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

$$f = \cos(u_1) + \exp(-u_2) \tag{11}$$

evaluated at full-grid input points where each dimension has k points. This function is broken down into fundamental operations:

$$\begin{aligned} \xi_1 &= \varphi_1(u_1) = \cos(u_1); \\ \xi_2 &= \varphi_2(u_2) = -u_2; \\ \xi_3 &= \varphi_3(\xi_2) = \exp(\xi_2); \\ f &= \varphi_4(\xi_1, \xi_3) = \xi_1 + \xi_3, \end{aligned} \tag{12}$$

and the accompanying computational graph is shown in Fig.1. The input points are given as

$$\boldsymbol{u} = \begin{Bmatrix} (u_1^{(1)}, u_2^{(1)}) & \dots & (u_1^{(k)}, u_2^{(1)}) \\ \vdots & \ddots & \vdots \\ (u_1^{(1)}, u_2^{(k)}) & \dots & (u_1^{(k)}, u_2^{(k)}) \end{Bmatrix}, \tag{13}$$

with k^2 input points. The computational graphs to evaluate this function with and without the AMTC method are shown in Fig. 2. 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 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)[31], 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, transforming the computational graph to minimize the required computational cost of evaluation a computational model on tensor-grid inputs.

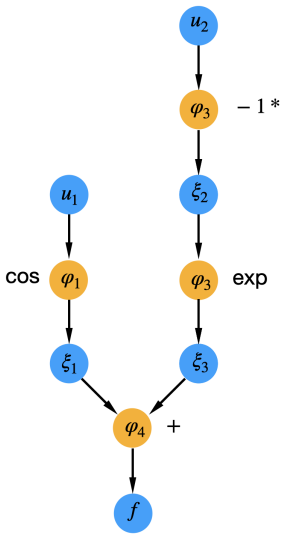
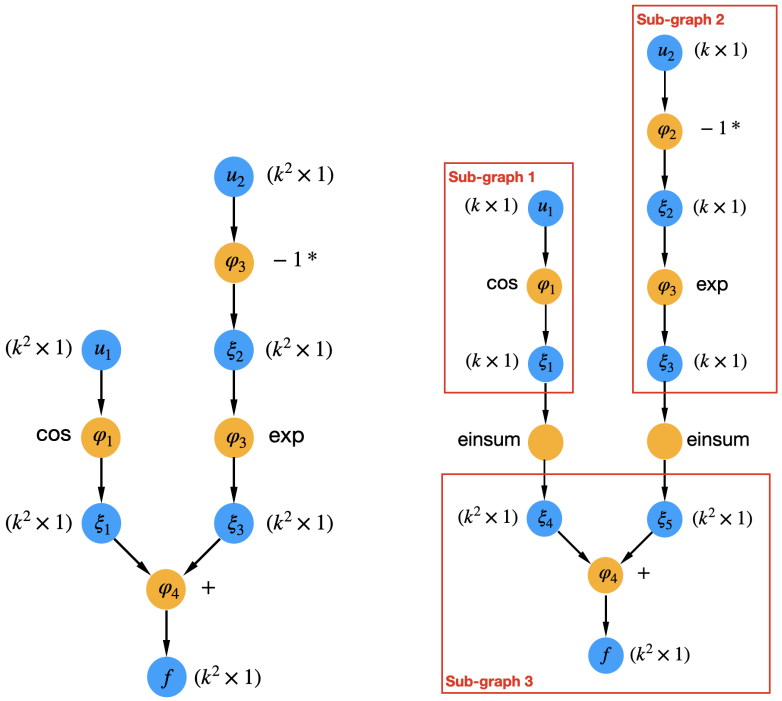


Fig. 1 Computational graph of function $f = cos(u_1) + exp(-u_2)$[27]



(a) Computational graph without using AMTC **(b) Computational graph using AMTC**

Fig. 2 Computational graphs with data size for full-grid input points evaluation on $f = cos(u_1) + exp(-u_2)$ [27]

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 using univariate gradient terms. With the addition of these gradient terms, the new approximation expression is more accurate when used to estimate the second and higher-order statistical moments while the required computational cost can be managed so that it scales linearly with the dimension of the random input space. We coin the new method as the gradient-enhanced univariate dimension reduction method (GUDR) method. In our method, we approximate the original function, f, as

$$\begin{aligned}\bar{f}(u) = &\sum_{i=1}^{d} f_i(u_i) - (d-1)f(\mu) + \sum_{i=1}^{d}(u_i - \mu_i)\sum_{j\neq i}^{d}\frac{\partial f_j}{\partial u_i}(u_j) \\ &- (d-1)\sum_{i}^{d}(u_i - \mu_i)\frac{\partial f}{\partial u_i}(\mu) - \sum_{i=1}^{d}\sum_{j>i}^{d}\frac{\partial^2 f}{\partial u_i \partial u_j}(\mu)(u_i - \mu_i)(u_j - \mu_j),\end{aligned} \tag{14}$$

where each term $f_i(u_i) = f(\mu_1, \ldots, \mu_{i-1}, u_i, \mu_{i+1}, \ldots, \mu_d)$ and $\frac{\partial f_j}{\partial u_i}(u_j) = \frac{\partial f}{\partial u_i}(\mu_1, \ldots, \mu_{j-1}, u_j, \mu_{j+1}, \ldots, \mu_d)$. This form includes all of the terms in the UDR approximation in (1), with additional terms including the univariate first-order gradient terms and one second-order term involving second derivatives of f. The GUDR approximation can also be

written in matrix form as

$$\bar{f}(x) = \sum_{i=1}^{d} f_i(u_i) - (d-1)f(\mu) + \sum_{i=1}^{d} \underbrace{\begin{bmatrix} u_1 - \mu_1 \\ \vdots \\ u_{i-1} - \mu_{i-1} \\ 0 \\ u_{i+1} - \mu_{i+1} \\ \vdots \\ u_d - \mu_d \end{bmatrix}^T}_{(u-\mu)\odot(\mathbf{1}-\mathbf{e_i})} \underbrace{\begin{bmatrix} \frac{\partial f_i}{\partial u_1}(u_i) \\ \vdots \\ \frac{\partial f_i}{\partial u_d}(u_i) \end{bmatrix}}_{\frac{\partial f_i}{\partial u}(u_i)}$$

$$- (d-1)\underbrace{\begin{bmatrix} u_1 - \mu_1 \\ \vdots \\ u_d - \mu_d \end{bmatrix}^T}_{u-\mu} \underbrace{\begin{bmatrix} \frac{\partial f}{\partial u_1}(\mu) \\ \vdots \\ \frac{\partial f}{\partial u_d}(\mu) \end{bmatrix}}_{\frac{\partial f}{\partial u}(\mu)} - \underbrace{\begin{bmatrix} u_1 - \mu_1 \\ \vdots \\ u_d - \mu_d \end{bmatrix}^T \begin{bmatrix} 0 & \frac{\partial^2 f}{\partial u_1 \partial u_2}(\mu) & \dots & \frac{\partial^2 f}{\partial u_1 \partial u_d}(\mu) \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & \frac{\partial^2 f}{\partial u_{d-1} \partial u_d}(\mu) \\ 0 & \dots & \dots & 0 \end{bmatrix} \begin{bmatrix} u_1 - \mu_1 \\ \vdots \\ u_d - \mu_d \end{bmatrix}}_{\frac{1}{2}(u-\mu)^T\left(\frac{\partial^2 f}{\partial u^2}(\mu) - \mathrm{diag}\left(\mathrm{diag}\left(\frac{\partial^2 f}{\partial u^2}(\mu)\right)\right)\right)(u-\mu)} \tag{15}$$

$$= \sum_{i=1}^{d} f_i(u_i) + \sum_{i=1}^{d} \left((u-\mu)\odot(\mathbf{1}-\mathbf{e_i})\right)^T \frac{\partial f_i}{\partial u}(u_i) - (d-1)\left(f(\mu) - (u-\mu)^T \frac{\partial f}{\partial u}(\mu)\right)$$

$$- \frac{1}{2}(u-\mu)^T \left(\frac{\partial^2 f}{\partial u^2}(\mu) - \mathrm{diag}\left(\mathrm{diag}\left(\frac{\partial^2 f}{\partial u^2}(\mu)\right)\right)\right)(u-\mu),$$

where, $\frac{\partial f}{\partial u}(u)$ and $\frac{\partial^2 f}{\partial u^2}(u)$ represents the gradient vector and Hessian matrix, respectively, evaluated at $u = u$. Compared to the Taylor series expansion of the original function, f, about u, the residual error of the GUDR approximation at $u = \mu$ can be expressed as

$$f(u) - \bar{f}(u) = \sum_{i=1}^{d}\sum_{j>i}^{d}\sum_{k>j}^{d} \frac{\partial^3 f}{\partial u_i \partial u_j \partial u_k}(\mu)(u_i - \mu_i)(u_j - \mu_j)(u_k - \mu_k) + \dots. \tag{16}$$

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 2nd 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

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

This means both UDR and GUDR approximations are expected to provide more accurate results in estimating the mean of the output than the 3rd-order Taylor series expansion method. 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

$$\mathbb{E}[f(U)^2] - \mathbb{E}[\bar{f}(U)^2] = \frac{1}{2!}f(\mu)\sum_{i_1=1}^{d-1}\sum_{i_2>i_1}^{d} \frac{\partial^4 f}{\partial u_{i_1}^2 \partial u_{i_2}^2}(\mu)\mathbb{E}[(u_{i_1} - \mu_{i_1})^2(u_{i_2} - \mu_{i_2})^2] + \dots \tag{18}$$

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 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 approximation, 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 the Appendix V.B.

B. Tensor-grid evaluations to estimate statistical moments

For a d-dimensional UQ problem, the GUDR approximation in (15) involves multiple one-dimensional function and first-order gradient evaluation 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,

$$\mathbb{E}[\bar{f}(U)] = \sum_{i=1}^{d} \int_{\Gamma_i} f_i(u_i)\rho(u_i)du_i - (d-1)f(\mu), \tag{19}$$

which involves the addition of multiple 1D integrals, which can be separately evaluated using quadrature rules. However, when it comes to estimating higher-order statistical moments of the output, there is no easy way to express the predictions of the GUDR approximation as a function of 1D integrals. 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).

If we treat each univariate function and gradient term in the GUDR approximation as individual operations, the computational graph of GUDR approximation function is shown in Fig.3. 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

$$\boldsymbol{u} = \boldsymbol{u}_1^k \times \ldots \times \boldsymbol{u}_d^k, \tag{20}$$

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 each operation of univariate function and gradient computation in the GUDR's computational graph is dependent on just one uncertain input, we can see that it is only necessary to evaluate these operations at the distinct quadrature points within their respective dimensions.

Consequently, by integrating the AMTC strategy, the modified computational graph for the GUDR approximation function's tensor-grid evaluations is shown in Fig. 4. 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 software and can only be applied to computational models that are built in the CSDL language. However, we also note that a large number of aircraft-design-relevant models have been developed across a wide range of disciplines [32]. Fortunately, in this specific scenario involving models implemented in CSDL, we can achieve the minimum number of evaluations in univariate function and gradient evaluation terms by first evaluating the univariate model function and gradient function on the corresponding quadrature points in its input space. Then we can 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, these data are passed into the GUDR approximation function to generate the full-grid evaluations of the output. We show the example code 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, these 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 other risk measures of the output like the probability of failure or the entire probability density distribution.

C. Computational cost

Utilizing the AMTC strategy enables the computation of any risk measure of the output with a fixed number of evaluations—specifically, 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 the advancement of 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 [33], 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 equivalent to $4kd + 3d + 4$ model evaluations, which scales only linearly with the dimension of the problem.

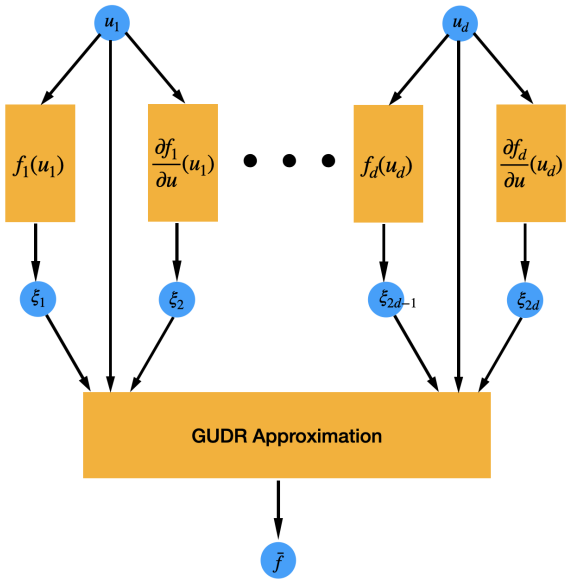


Fig. 3 Computational graph of the GUDR approximation function

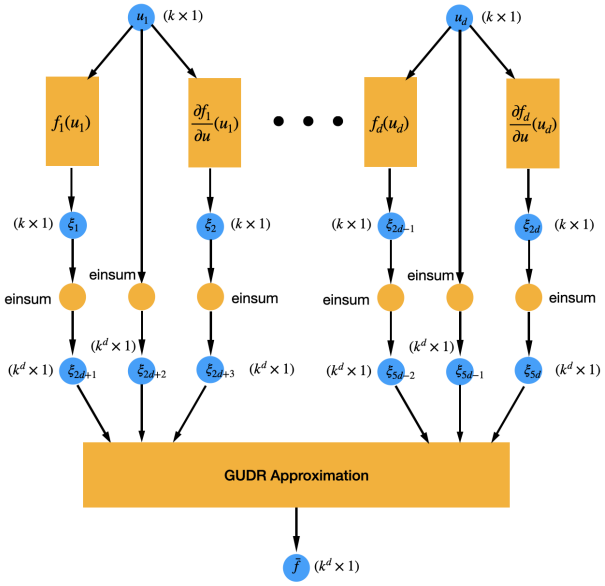


Fig. 4 Computational graph of tensor-grid evaluations after graph transformation

IV. Numerical Results

In Sec IV.A, we compare the accuracy of four UQ methods: gradient-enhanced univariate dimension reduction (GUDR), univariate dimension reduction (UDR), method of moments using 2nd order Taylor expansion, and method of moments using 3rd order Taylor expansion. These methods are applied to two UQ problems involving different mathematical functions. In Sec IV.B, we compare the convergence plots of GUDR, UDR, and the Monte Carlo methods on a 7D UQ problem involving an aircraft design multidisciplinary model.

A. Mathematical functions

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

$$y_1 = f(x_1, x_2) = \frac{1}{1 + x_1^4 + 2x_2^2 + x_2^4} \tag{21}$$

with uncertain inputs:

$$X_i \sim \mathcal{N}(2, \sigma), \quad i = 1, 2. \tag{22}$$

The second problem involves the function:

$$y_2 = f(x_1, x_2, x_3) = \exp(1 + 0.5x_1^2 + 0.5x_2^2 + 0.5x_3^2) \tag{23}$$

with uncertain inputs:

$$X_i \sim \mathcal{N}(3, \sigma), \quad i = 1, 2, 3. \tag{24}$$

The quantity of interest (QoI) for both UQ problems is the standard deviations σ_1 and σ_2 of the outputs Y_1 and Y_2, respectively. These problems are adapted from [22].

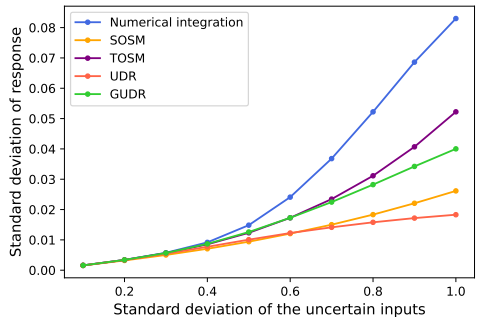
For these test problems, we have implemented five methods: UDR, GUDR, method of moments using 2nd-order Taylor series expansion (second-order second-moment (SOSM)), method of moments using 3rd-order Taylor series expansion (third-order second-moment (TOSM)), and direct numerical integration. The UDR 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 true UQ results.

The UQ outcomes derived from these four methods, along with the relative errors of the three analytical expression methods, are plotted against the increasing values of input standard deviations in Fig. 5 and Fig. 6 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 between GUDR and TOSM match our theoretical results showing that GUDR approximation function is expected to have comparable performance as the 3rd-order Taylor expansion when estimating the standard deviation of the output.

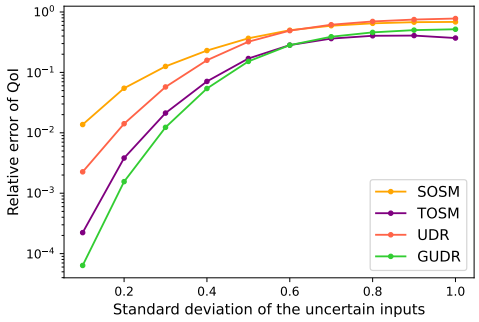
B. Aircraft design multidisciplinary model

The third 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 high altitude while receiving charge from a laser beam emitted by a ground station. The concept of the UAV is illustrated in Fig. 7, while the laser beam engagement scenario is depicted in Fig. 8. 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 the aerodynamics and structure solvers are shown in Fig. 9. For further details on the computational model, we refer the reader to [34].

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. 1.

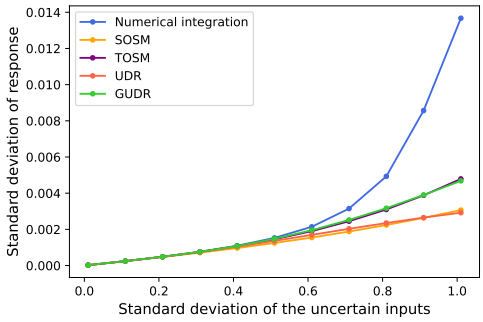


(a) QoI results for increasing values of input standard deviation

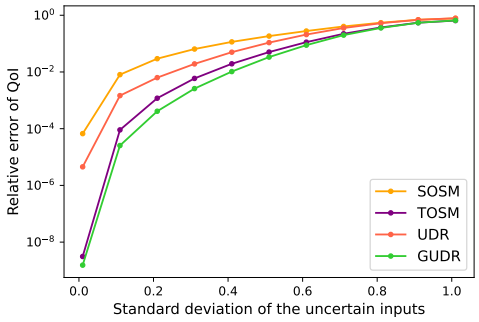


(b) Relative errors for increasing values of input standard deviation

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



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



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

Fig. 6 UQ results for $y_2 = e^{(1+0.5x_1^2+0.5x_2^2+0.5x_3^2)}$

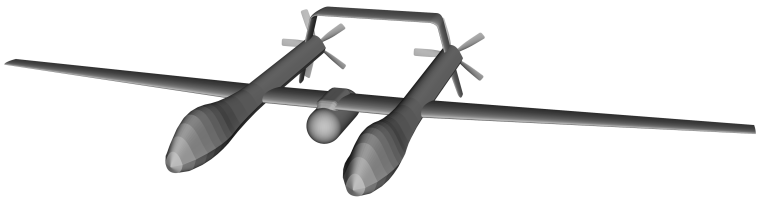


Fig. 7 The high-altitude, laser-powered airplane concept

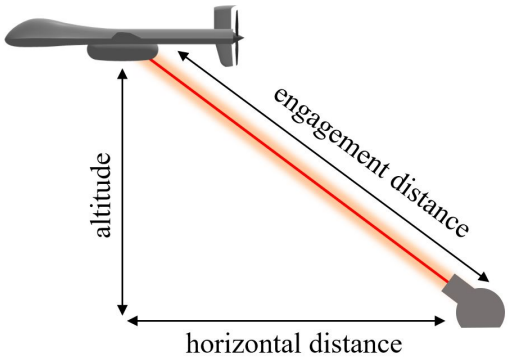
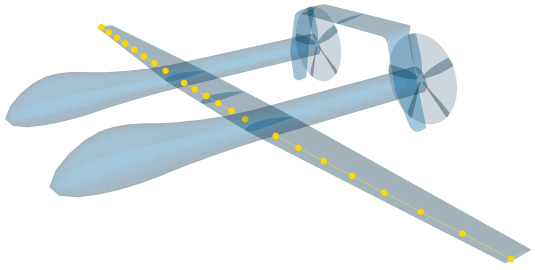


Fig. 8 The laser beam engagement scenario

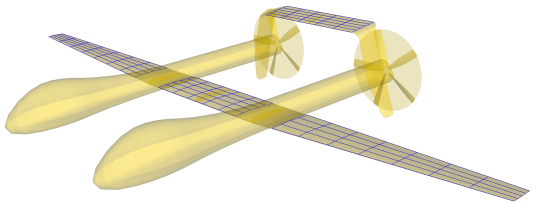
Uncertain inputs	Distributions
Cruise mach number	$U(0.1, 0.2)$
Altitude(km)	$U(17, 19)$
Horizontal distance (km)	$U(250, 300)$
Pitch angle (deg)	$U(-2, 2)$
Payload mass (kg)	$U(50, 100)$
Aperture diameter (m)	$U(0.7, 0.8)$
Rotor speed (RPM)	$U(1200, 1500)$

Table 1 Input parameters and ranges for the multidisciplinary model

In this test problem, we implemented three UQ methods: GUDR, UDR, and the Monte Carlo method. The GUDR and UDR methods were implemented with varying numbers of quadrature points in each dimension, namely

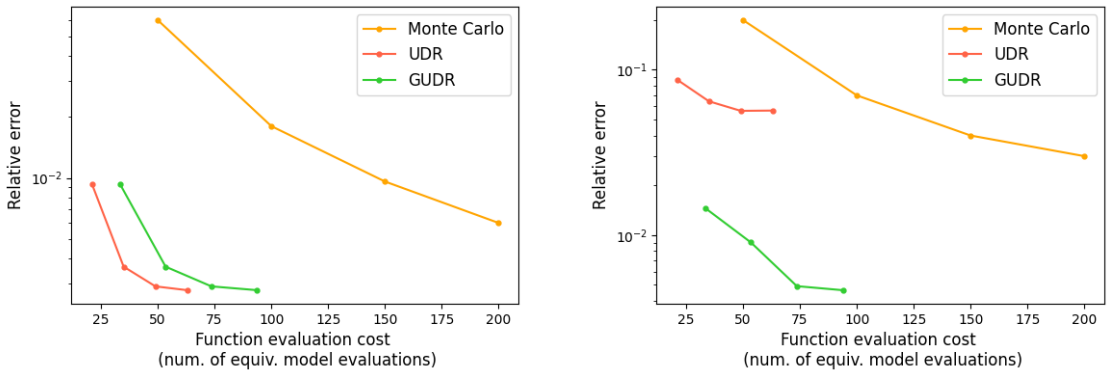


(a) 1D beam meshes for the structure solver



(b) 2D aerodynamics meshes for the aerodynamics solver

Fig. 9 Meshes for structure and aerodynamics solvers



(a) Mean of the output **(b) Standard deviation of the output**

Fig. 10 Convergence plots for the aircraft design problem

$n = 3, 5, 7, 9$. On the other hand, the Monte Carlo method was implemented with different numbers of sample points, $m = 50, 100, 150, 200$. The computational model for this problem was implemented in the Computational System Design Language (CSDL) [31]. The first-order gradient evaluations required for GUDR were calculated using an automatically implemented adjoint method detailed in [35], and the Hessian matrix was estimated via a finite difference method. We used the UQ results generated from the Monte Carlo method with $10,000$ sample points as the reference results and depicted the convergence plots of the three methods in Fig. 10. The relative errors for the Monte Carlo were computed as the average over 20 runs.

From the results, it is evident that at a low evaluation cost (10s of equivalent model evaluations), both UDR and GUDR 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 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 derivations, which affirm that both GUDR and UDR are fourth-order accurate and produce identical results when estimating the mean of the output. As for the standard deviation of the output, GUDR enhances the accuracy of UDR by almost an order of magnitude at the expense of a slightly increased function evaluation 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 50 equivalent model evaluations, GUDR can achieve a relative error of less than 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.

V. Conclusion

In this paper, we introduce 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) by incorporating 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 risk measure of the output. By using an automatic differentiation method to evaluate the gradient information, GUDR preserves the linear scaling cost as the UDR method.

The GUDR method was tested on three UQ problems: two mathematical functions in 2D and 3D, and a 7D aircraft design problem. For the mathematical functions, GUDR surpassed UDR in accuracy for standard deviation estimation and matched the performance of the method of moments using 3rd-order Taylor expansion. In the complex 7D aircraft design problem, GUDR improved upon UDR's precision by tenfold when estimating the output's standard deviation, achieving less than 1% relative error for both mean and standard deviation estimations, with a computational expense equivalent to only 50 model evaluations.

One 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. A

promising direction for future work is to assess the GUDR's accuracy in estimating reliability metrics of the output by connecting GUDR with polynomial chaos and kriging methods. Another interesting topic is to propose more accurate linear-scaling UQ methods following the GUDR idea.

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 $f\left(u_1, u_2\right)$, the Taylor series expansion of f at $(u_1 = \mu_1, u_2 = \mu_2)$ can be expressed by

$$\begin{aligned} f\left(u_1, u_2\right) =& f(\mu_1, \mu_2) + \frac{\partial f}{\partial x_1}(\mu_1, \mu_2)(u_1 - \mu_1) + \frac{\partial f}{\partial u_2}(\mu_1, \mu_2)(u_2 - \mu_2) + \frac{1}{2!}\frac{\partial^2 f}{\partial u_1^2}(\mu_1, \mu_2)(u_1 - \mu_1)^2 \\ &+ \frac{1}{2!}\frac{\partial^2 f}{\partial u_2^2}(\mu_1, \mu_2)(u_2 - \mu_2)^2 + \frac{\partial^2 f}{\partial u_1 \partial u_2}(\mu_1, \mu_2)(u_1 - \mu_1)(u_2 - \mu_2) + \frac{1}{3!}\frac{\partial^3 f}{\partial u_1^3}(\mu_1, \mu_2)(u_1 - \mu_1)^3 \\ &+ \frac{1}{3!}\frac{\partial^3 f}{\partial u_2^3}(\mu_1, \mu_2)(u_2 - \mu_2)^3 + \frac{1}{2!}\frac{\partial^3 f}{\partial u_1^2 \partial u_2}(\mu_1, \mu_2)(u_1 - \mu_1)^2(u_2 - \mu_2) \\ &+ \frac{1}{2!}\frac{\partial^3 f}{\partial u_1 \partial u_2^2}(\mu_1, \mu_2)(u_1 - \mu_1)(u_2 - \mu_2)^2 + \frac{1}{4!}\frac{\partial^4 f}{\partial u_1^4}(\mu_1, \mu_2)(u_1 - \mu_1)^4 + \frac{1}{4!}\frac{\partial^4 f}{\partial u_2^4}(\mu_1, \mu_2)(u_2 - \mu_2)^4 \\ &+ \frac{1}{3!}\frac{\partial^4 f}{\partial u_1^3 \partial u_2}(\mu_1, \mu_2)(u_1 - \mu_1)^3(u_2 - \mu_2) + \frac{1}{3!}\frac{\partial^4 f}{\partial u_1 \partial u_2^3}(\mu_1, \mu_2)(u_1 - \mu_1)(u_2 - \mu_2)^3 \\ &+ \frac{1}{2!2!}\frac{\partial^4 f}{\partial u_1^2 \partial u_2^2}(\mu_1, \mu_2)(u_1 - \mu_1)^2(u_2 - \mu_2)^2 + \ldots \end{aligned} \tag{25}$$

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

$$\hat{f}(u_1, u_2) = f(u_1, \mu_2) + f(\mu_1, u_2) - f(\mu_1, \mu_2). \tag{26}$$

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

$$\begin{aligned} \hat{f}(u_1, u_2) =& f(\mu_1, \mu_2) + \frac{\partial f}{\partial u_1}(\mu_1, \mu_2)(u_1 - \mu_1) + \frac{\partial f}{\partial u_2}(\mu_1, \mu_2)(u_2 - \mu_2) + \frac{1}{2!}\frac{\partial^2 f}{\partial u_1^2}(\mu_1, \mu_2)(u_1 - \mu_1)^2 \\ &+ \frac{1}{2!}\frac{\partial^2 f}{\partial u_2^2}(\mu_1, \mu_2)(u_2 - \mu_2)^2 + \frac{1}{3!}\frac{\partial^3 f}{\partial u_1^3}(\mu_1, \mu_2)(u_1 - \mu_1)^3 + \frac{1}{3!}\frac{\partial^3 f}{\partial u_2^3}(\mu_1, \mu_2)(u_2 - \mu_2)^3 \\ &+ \frac{1}{4!}\frac{\partial^4 f}{\partial u_1^4}(\mu_1, \mu_2)(x_1 - \mu_1)^4 + \frac{1}{4!}\frac{\partial^4 f}{\partial x_2^4}(\mu_1, \mu_2)(u_2 - \mu_2)^4 + \ldots \end{aligned} \tag{27}$$

If we compare the Taylor series expansion of the original function with the UDR approximation, all of the Taylor series terms in (27) are contained in (25), and the residual errors can be expressed as

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

From (28), 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

$$\begin{aligned}\mathbb{E}[f(U_1,U_2)]-\mathbb{E}[\hat{f}(U_1,U_2)] &= \int_{\Gamma_1}\int_{\Gamma_2}\left(f(u_1,u_2)-\hat{f}(u_1,u_2)\right)\rho(u_1)\rho(u_2)du_1du_2 \\ &= \frac{1}{2!2!}\frac{\partial^4 f}{\partial u_1^2\partial u_2^2}(\mu_1,\mu_2)\mathbb{E}[(u_1-\mu_1)^2(u_2-\mu_2)^2]+\dots\end{aligned} \tag{29}$$

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

$$\begin{aligned}\tilde{f}\left(u_1,u_2\right) =& f(\mu_1,\mu_2)+\frac{\partial f}{\partial x_1}(\mu_1,\mu_2)(u_1-\mu_1)+\frac{\partial f}{\partial u_2}(\mu_1,\mu_2)(u_2-\mu_2)+\frac{1}{2!}\frac{\partial^2 f}{\partial u_1^2}(\mu_1,\mu_2)(u_1-\mu_1)^2 \\ &+\frac{1}{2!}\frac{\partial^2 f}{\partial u_2^2}(\mu_1,\mu_2)(u_2-\mu_2)^2+\frac{\partial^2 f}{\partial u_1\partial u_2}(\mu_1,\mu_2)(u_1-\mu_1)(u_2-\mu_2)+\frac{1}{3!}\frac{\partial^3 f}{\partial u_1^3}(\mu_1,\mu_2)(u_1-\mu_1)^3 \\ &+\frac{1}{3!}\frac{\partial^3 f}{\partial u_2^3}(\mu_1,\mu_2)(u_2-\mu_2)^3+\frac{1}{2!}\frac{\partial^3 f}{\partial u_1^2\partial u_2}(\mu_1,\mu_2)(u_1-\mu_1)^2(u_2-\mu_2) \\ &+\frac{1}{2!}\frac{\partial^3 f}{\partial u_1\partial u_2^2}(\mu_1,\mu_2)(u_1-\mu_1)(u_2-\mu_2)^2\end{aligned} \tag{30}$$

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

$$\begin{aligned}\mathbb{E}[f(U_1,U_2)]-\mathbb{E}[\tilde{f}(U_1,U_2)] =& \frac{1}{2!2!}\frac{\partial^4 f}{\partial u_1^2\partial u_2^2}(\mu_1,\mu_2)\mathbb{E}[(u_1-\mu_1)^2(u_2-\mu_2)^2]+\frac{1}{4!}\frac{\partial^4 f}{\partial u_1^4}(\mu_1,\mu_2)\mathbb{E}[(u_1-\mu_1)^4] \\ &+\frac{1}{4!}\frac{\partial^4 f}{\partial u_2^4}(\mu_1,\mu_2)\mathbb{E}[(u_2-\mu_2)^4]+\dots\end{aligned} \tag{31}$$

Comparing the residual errors of the 3rd-order Taylor series expansion method in (31) with the residual errors of the UDR in (29). 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:

$$\begin{aligned}\mathbb{E}[f(U_1,U_2)^2]-\mathbb{E}[\hat{f}(U_1,U_2)^2] &= \int_{\Gamma_1}\int_{\Gamma_2}\left(f(u_1,u_2)^2-\hat{f}(u_1,u_2)^2\right)\rho(u_1)\rho(u_2)du_1du_2 \\ &= \left(\frac{\partial^2 f}{\partial u_1\partial u_2}(\mu_1,\mu_2)\right)^2\mathbb{E}[(u_1-\mu_1)^2(u_2-\mu_2)^2] \\ &+\left(\frac{\partial y}{\partial u_2}(\mu_1,\mu_2)\frac{\partial^3 y}{\partial u_1^2\partial u_2}(\mu_1,\mu_2)+\frac{\partial y}{\partial u_1}(\mu_1,\mu_2)\frac{\partial^3 y}{\partial u_1\partial u_2^2}(\mu_1,\mu_2)\right)\mathbb{E}[(u_1-\mu_1)^2(u_2-\mu_2)^2] \\ &+\frac{1}{2!}f(\mu_1,\mu_2)\frac{\partial^4 f}{\partial u_1^2\partial u_2^2}(\mu_1,\mu_2)\mathbb{E}[(u_1-\mu_1)^2(u_2-\mu_2)^2]+\dots\end{aligned} \tag{32}$$

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

$$\mathbb{E}[f(U_1,U_2)^2]-\mathbb{E}[\tilde{f}(U_1,U_2)^2]=\frac{1}{2!}f(\mu_1,\mu_2)\frac{\partial^4 f}{\partial u_1^2\partial u_2^2}(\mu_1,\mu_2)\mathbb{E}[(u_1-\mu_1)^2(u_2-\mu_2)^2]+\dots \tag{33}$$

Comparing the residual errors of the 3rd-order Taylor series expansion method in (40) with the residual errors of the UDR in (32). 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 $f\left(u_1, u_2\right)$, the GUDR method approximates the original function as

$$\begin{aligned}\bar{f}(u_1,u_2) = y(u_1,\mu_2) + y(\mu_1,u_2) - y(\mu_1,\mu_2) + (u_1-\mu_1)\left(\frac{\partial f}{\partial u_1}(\mu_1,u_2) - \frac{\partial f}{\partial u_1}(\mu_1,\mu_2)\right) \\ + (u_2-\mu_2)\left(\frac{\partial f}{\partial u_2}(u_1,\mu_2) - \frac{\partial f}{\partial u_2}(\mu_1,\mu_2)\right) - \frac{\partial^2 f}{\partial u_1 u_2}(\mu_1,\mu_2)(x_1-\mu_1)(x_2-\mu_2).\end{aligned} \tag{34}$$

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

$$\frac{\partial f}{\partial u_1}(\mu_1,u_2) = \frac{\partial f}{\partial u_1}(\mu_1,\mu_2) + \frac{\partial^2 f}{\partial u_1 u_2}(\mu_1,\mu_2)(u_2-\mu_2) + \frac{1}{2!}\frac{\partial^3 f}{\partial u_1 u_2^2}(\mu_1,\mu_2)(u_2-\mu_2)^2 + \frac{1}{3!}\frac{\partial^4 f}{\partial u_1 u_2^3}(\mu_1,\mu_2)(u_2-\mu_2)^3 + \dots \tag{35}$$

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

Now we can express the Taylor series expansion of GUDR approximation function, $\bar{f}$, as

$$\begin{aligned}\bar{f}\left(u_1,u_2\right) =& f(\mu_1,\mu_2) + \frac{\partial f}{\partial u_1}(\mu_1,\mu_2)(u_1-\mu_1) + \frac{\partial f}{\partial u_2}(\mu_1,\mu_2)(u_2-\mu_2) + \frac{1}{2!}\frac{\partial^2 f}{\partial u_1^2}(\mu_1,\mu_2)(u_1-\mu_1)^2 \\ &+ \frac{1}{2!}\frac{\partial^2 f}{\partial u_2^2}(\mu_1,\mu_2)(u_2-\mu_2)^2 + \frac{\partial^2 f}{\partial u_1 \partial u_2}(\mu_1,\mu_2)(u_1-\mu_1)(u_2-\mu_2) + \frac{1}{3!}\frac{\partial^3 f}{\partial u_1^3}(\mu_1,\mu_2)(u_1-\mu_1)^3 \\ &+ \frac{1}{3!}\frac{\partial^3 f}{\partial u_2^3}(\mu_1,\mu_2)(u_2-\mu_2)^3 + \frac{1}{2!}\frac{\partial^3 f}{\partial u_1^2 \partial u_2}(\mu_1,\mu_2)(u_1-\mu_1)^2(u_2-\mu_2) \\ &+ \frac{1}{2!}\frac{\partial^3 f}{\partial u_1 \partial u_2^2}(\mu_1,\mu_2)(u_1-\mu_1)(u_2-\mu_2)^2 + \frac{1}{4!}\frac{\partial^4 f}{\partial u_1^4}(\mu_1,\mu_2)(u_1-\mu_1)^4 + \frac{1}{4!}\frac{\partial^4 f}{\partial u_2^4}(\mu_1,\mu_2)(u_2-\mu_2)^4 \\ &+ \frac{1}{3!}\frac{\partial^4 f}{\partial u_1^3 \partial u_2}(\mu_1,\mu_2)(u_1-\mu_1)^3(u_2-\mu_2) + \frac{1}{3!}\frac{\partial^4 f}{\partial u_1 \partial u_2^3}(\mu_1,\mu_2)(u_1-\mu_1)(u_2-\mu_2)^3 + \dots\end{aligned} \tag{37}$$

If we compare the Taylor series expansion of the original function with the GUDR approximation, all of the Taylor series terms in (37) are contained in (25), and the residual errors can be expressed as

$$f\left(u_1,u_2\right) - \bar{f}\left(u_1,u_2\right) = \frac{1}{2!2!}\frac{\partial^4 f}{\partial u_1^2 \partial u_2^2}(0,0)u_1^2 u_2^2 + \dots \tag{38}$$

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

$$\mathbb{E}[f(u_1,u_2)] - \mathbb{E}[\bar{f}(u_1,u_2)] = \frac{1}{2!2!}\frac{\partial^4 f}{\partial u_1^2 \partial u_2^2}(\mu_1,\mu_2)\mathbb{E}[(u_1-\mu_1)^2(u_2-\mu_2)^2] + \dots \tag{39}$$

However when we estimate the residual errors, the residual errors for GUDR approximation can be expressed as:

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

Compared to the 3rd-order Taylor series expansion, 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.

```
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('i...,p...->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]])
  a2 = np.array([u_1[i], 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, 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 evaluated on tensor-grid quadrature points, size (k^2,)
```


Acknowledgments

The material presented in this paper is, in part, based upon work supported by DARPA under grant No. D23AP00028-00

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] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] Piric, K., "Reliability analysis method based on determination of the performance function's 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.

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

[25] 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.

[26] 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.

[27] Wang, B., Sperry, M., Gandarillas, V. E., and Hwang, J. T., "Accelerating model evaluations in uncertainty propagation on tensor grids using computational graph transformations," *arXiv preprint arXiv:2309.06617*, 2023. https://doi.org/10.48550/arXiv.2309.06617.

[28] 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.

[29] Wang, B., Orndorff, N. C., Sperry, M., and Hwang, J. T., "High-dimensional uncertainty quantification using graph-accelerated non-intrusive polynomial chaos and active subspace methods," *AIAA AVIATION 2023 Forum*, 2023, p. 4264. https://doi.org/10.2514/6.2023-4264.

[30] Wang, B., Orndorff, N. C., and Hwang, J. T., "Optimally tensor-structured quadrature rule for uncertainty quantification," *AIAA SCITECH 2023 Forum*, 2023, p. 0741. https://doi.org/10.2514/6.2023-0741.

[31] 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*, under revision.

[32] Sarojini, D., Ruh, M. L., Joshy, A. J., Yan, J., Ivanov, A. K., Scotzniovsky, L., Fletcher, A. H., Orndorff, N. C., Sperry, M., Gandarillas, V. E., et al., "Large-Scale Multidisciplinary Design Optimization of an eVTOL Aircraft using Comprehensive Analysis," *AIAA SCITECH 2023 Forum*, 2023, p. 0146. https://doi.org/10.2514/6.2023-0146.

[33] 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.

[34] 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.

[35] 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.