This is the preprint version of the following article:
Bingran Wang, Nicholas C. Orndorff, and John T. Hwang. Graph-accelerated non-intrusive polynomial chaos expansion using partially tensor-structured quadrature rules for uncertainty quantification, Aerospace Science and Technology, vol. 155, 2024, 109607.

Published article: https://doi.org/10.1016/j.ast.2024.109607
Preprint pdf: https://github.com/LSDOlab/lsdo_bib/blob/main/pdf/wang2024partial.pdf
Bibtex: https://github.com/LSDOlab/lsdo_bib/blob/main/bib/wang2024partial.bib

Graph-accelerated non-intrusive polynomial chaos expansion using partially tensor-structured quadrature rules for uncertainty quantification

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

Recently, the graph-accelerated non-intrusive polynomial chaos (NIPC) method has been proposed for solving uncertainty quantification (UQ) problems. This method leverages the full-grid integration-based NIPC method to address UQ problems while employing the computational graph transformation approach, AMTC, to accelerate the tensor-grid evaluations. This method exhibits remarkable efficacy on a broad range of low-dimensional (three dimensions or less) UQ problems featuring multidisciplinary models. However, it often does not scale well with problem dimensions due to the exponential increase in the number of quadrature points when using the full-grid quadrature rule. To expand the applicability of this method to a broader range of UQ problems, this paper introduces a new framework for generating a tailored, partially tensor-structured quadrature rule to use with the graph-accelerated NIPC method. This quadrature rule, generated through the designed quadrature approach, possesses a tensor structure that is tailored for the computational model. The selection of the tensor structure is guided by an analysis of the computational graph, ensuring that the quadrature rule effectively capitalizes on the sparsity within the computational graph when paired with the AMTC method. This method has been tested on one 4D and one 6D UQ problem, both originating from aircraft design scenarios and featuring multidisciplinary models. Numerical results show that, when using with graph-accelerated NIPC method, our approach generates a partially tensor-structured quadrature rule that outperforms the full-grid Gauss quadrature and the designed quadrature methods (more than 40% reduction in computational costs) in both of the test problems.

I. Introduction

Engineers rely heavily on computational models to design, analyze, and optimize engineering systems. However, such computational models are often deterministic and inevitably affected by uncertainties inherent in real-world scenarios. Uncertainty quantification (UQ) plays a pivotal role in ensuring the reliability of predictions made by these computational models. Its applications span various scientific and engineering domains, including weather forecasting [1, 2], machine learning [3, 4]), structural analysis [5, 6], and aircraft design [7–9], where uncertainties can significantly impact system behavior and performance.

Forward UQ, also known as uncertainty propagation, seeks to assess how uncertainties associated with model inputs affect a computational model's outputs or quantities of interest (QoIs). These input uncertainties can stem from variations in operating conditions and model parameters, referred to as aleatoric uncertainties, or from model errors due to incomplete knowledge, termed epistemic uncertainties. By quantifying the impact of input uncertainties, forward UQ plays a crucial role in supporting decision-making and risk assessment in engineering system design. In this paper, we address a UQ problem within a probabilistic framework, representing uncertain inputs as continuous probability density distributions. The objective is to estimate the statistical moments or risk measures of the QoI.

We focus on employing the non-intrusive polynomial chaos (NIPC) method [10] to solve the UQ problem, a popular approach especially suitable for UQ problems with fewer than 10 dimensions. The NIPC method approximates the model response through a linear combination of polynomial chaos expansion (PCE) basis functions, determined by the distributions of the random inputs. The PCE coefficients are computed using either sampling methods [11–13] or integration methods [14–16]. In [17], Eldred and Burkardt demonstrated that for low-dimensional UQ problems, the integration-based and regression-based NIPC methods show comparable performance, with significantly higher convergence rates than the Monte Carlo method.

The integration approach makes use of the orthogonality properties of the PCE basis functions and transforms the UQ problems into multidimensional integration problems. Common methods for solving the integration problems include Gauss quadrature [18], Smolyak sparse-grid [19], and designed quadrature [20] methods. The Gauss quadrature method formulates the quadrature rule using a tensor product of univariate quadrature rules. This approach is easy to use but the number of quadrature points increases exponentially with the number of dimensions. The Smolyak sparse-grid approach breaks the tensor structure of the Gauss quadrature rule by dropping the higher-order cross terms while achieving minimal sacrifice on its accuracy. In comparison, the design quadrature method generates the quadrature rule by solving an optimization problem to find approximate solutions to the moment-matching equations. The designed quadrature method results in a quadrature rule that does not possess a tensorial structure but can typically achieve better performance than the previous two methods.

Wang et al. recently introduced a new method called *Accelerated Model evaluations on Tensor grids using Computational graph transformations* (AMTC) [21]. This method reduces the computational cost of model evaluations on tensor-grid inputs by modifying the computational graph of the model within the middle end of a modeling language's three-stage compiler. This modification eliminates redundant evaluations at the operation level caused by the tensor structure of the inputs. Their previous work applies AMTC with integration-based NIPC to solve UQ problems, under the assumption of a specific type of sparsity within the computational graphs of the computational models. This integration is termed the *graph-accelerated NIPC* method and has demonstrated superior efficiency in various UQ and optimization under uncertainty problems involving multidisciplinary systems [8]. However, the graph-accelerated NIPC method has so far only been implemented with a full-grid Gauss quadrature rule, targeting low-dimensional UQ problems.

The primary contribution of our paper is a new framework to generate partially tensor-structured quadrature rules that can be used with the graph-accelerated NIPC methods to solve UQ problems with multidisciplinary models. The proposed method generates the quadrature rule following using a particular tensor structure such that the associated model evaluations can be accelerated using the recently proposed computational graph acceleration method, AMTC. Within this framework, we first identify a suitable tensor structure option for the quadrature rule through an analysis of the computational model. Subsequently, we employ the designed quadrature method to construct the quadrature rule that adheres to this tensor structure. The resulting quadrature rule is strategically designed to leverage the sparsity inherent in the computational graph of the model, thus maximizing efficiency when employing the AMTC method for accelerating tensor-grid evaluations.

This method has been applied to two UQ problems, one with four uncertain inputs and one with six uncertain inputs, both derived from practical aircraft design problems involving multidisciplinary systems. The proposed method produces new partially tensor-structured quadrature rules distinct from existing ones for both test problems. These partially structured quadrature rules also outperform existing methods when employed with the graph-accelerated NIPC method for solving UQ problems.

This paper is structured as follows. Section II provides background information on NIPC, quadrature rules, and AMTC. In Section III, we elaborate on the proposed framework. Section IV presents the numerical results obtained from the test problems. Finally, Section V summarizes the work and provides concluding remarks.

II. Background

A. Non-intrusive polynomial chaos

Polynomial chaos, also known as polynomial chaos expansion (PCE), traces its origins back to Wiener, who utilized Hermite polynomials in developing a theoretical framework for Gaussian random processes [22]. Ghanem further expanded upon this concept, employing Hermite polynomials as orthogonal bases to represent random processes [23]. However, this method suffers from convergence issues for problems with non-Gaussian uncertain inputs. This issue was addressed by Xiu and Karniadakis in their work [10], wherein they introduced the generalized polynomial chaos

(gPC) method. The gPC method strategically employs different orthogonal polynomials tailored to uncertain inputs with different distributions, thereby achieving optimal convergence in addressing UQ problems.

We consider a UQ problem involving a function represented as

$$f = \mathcal{F}(u), \tag{1}$$

where $\mathcal{F} : \mathbb{R}^d \to \mathbb{R}$ represents a deterministic model evaluation function, $u \in \mathbb{R}^d$ denotes the input vector, and $f \in \mathbb{R}$ represents a scalar output. The UQ problem aims to compute the statistical moments or risk measures of the stochastic $f(U)$ under the effect of the input uncertainties, denoted as a stochastic vector, $U := [U_1, \ldots, U_d]$. The input uncertainties are assumed to be mutually independent and adhere to the probability distribution $\rho(u)$ with support Γ.

Using the gPC method, the stochastic output can be represented as a linear combination of orthogonal polynomials:

$$f(U) = \sum_{i=0}^{\infty} \alpha_i \Phi_i(U), \tag{2}$$

where $\Phi_i(U)$ denotes the PCE basis functions that are chosen based on the uncertain inputs' probability distribution $\rho(u)$, and α_i represents the PCE coefficients that need to be determined. In practical applications, it is common to truncate the PCE basis functions up to a specific polynomial order and approximate the stochastic output as

$$f(U) \approx \sum_{i=0}^{q} \alpha_i \Phi_i(U). \tag{3}$$

With p as the upper bound for the total polynomial order, the resultant number of PCE basis functions, $q + 1$, satisfies

$$q + 1 = \frac{(d + p)!}{d!p!}. \tag{4}$$

The PCE basis functions are chosen to satisfy the orthogonality property as

$$\left\langle \Phi_i(U), \Phi_j(U) \right\rangle = \delta_{ij}, \tag{5}$$

where δ_{ij} is the Kronecker delta and the inner product is defined as

$$\left\langle \Phi_i(U), \Phi_j(U) \right\rangle = \int_{\Gamma} \Phi_i(u)\Phi_j(u)\rho(u)du. \tag{6}$$

In the one-dimensional case, the PCE basis functions consist of univariate orthogonal polynomials chosen according to the distribution of the uncertain input [10]. Tab. 1 presents common types of continuous random variables alongside their corresponding orthogonal polynomials and support ranges. In multi-dimensional cases, the PCE basis functions are formed as tensor products of the univariate orthogonal polynomials. For instance, in the 2D case, the first six PCE basis functions are expressed as

$$\begin{aligned}
\Phi_0(U) &= \phi_0(U_1)\phi_0(U_2) \\
\Phi_1(U) &= \phi_1(U_1)\phi_0(U_2) \\
\Phi_2(U) &= \phi_0(U_1)\phi_1(U_2) \\
\Phi_3(U) &= \phi_2(U_1)\phi_0(U_2) \\
\Phi_4(U) &= \phi_1(U_1)\phi_1(U_2) \\
\Phi_5(U) &= \phi_0(U_1)\phi_2(U_2),
\end{aligned}$$

where $\phi_i(U_j)$ represents the i-th univariate orthogonal polynomial corresponding to the uncertain input, U_j. If the uncertain inputs are dependent, the PCE basis functions can be generated numerically following the approach outlined in [24].

Using the gPC approximation in 3, the statistical moments of the QoI can be analytically computed as

$$\mu_f = \alpha_0, \tag{7}$$

$$\sigma_f = \sum_{i=i}^{d} \alpha_i^2 \left\langle \Phi_i^2 \right\rangle. \tag{8}$$

Table 1 Orthogonal polynomials for common types of continuous random variables

Distribution	Orthogonal polynomials	Support range
Normal	Hermite	$(-\infty, \infty)$
Uniform	Legendre	$[-1, 1]$
Exponential	Laguerre	$[0, \infty)$
Beta	Jacobi	$(-1, 1)$
Gamma	Generalized Laguerre	$[0, \infty)$

For risk measures like probability of failure or Conditional Value at Risk (CVaR), the gPC model can be used as an inexpensive-to-evaluate surrogate model, and one can apply sampling methods to approximate them.

To solve the PCE coefficients α_i in (3), the non-intrusive polynomial chaos (NIPC) method applies either the integration or regression approach. The integration approach makes use of the orthogonality property in (5) and projects the QoI onto each basis function, which results in a multi-dimensional integration problem:

$$\begin{aligned}\alpha_i &= \frac{\left\langle f(U), \Phi_i \right\rangle}{\left\langle \Phi_i^2 \right\rangle} \\ &= \frac{1}{\left\langle \Phi_i^2 \right\rangle} \int_\Gamma f(u)\Phi_i(u)\rho(u)du \\ &= \frac{1}{\left\langle \Phi_i^2 \right\rangle} \int_{\Gamma_1} \cdots \int_{\Gamma_d} f(u_1, \ldots, u_d)\Phi_i(u_1, \ldots, u_d)\rho(u_1, \ldots, u_d)du_1 \ldots du_d.\end{aligned} \tag{9}$$

In practical problems, this integration problem is often approximated using numerical quadrature rules.

B. Numerical quadrature rules

This paper focuses on employing the integration-based NIPC method for solving multi-dimensional UQ problems. In this context, the integration-based NIPC method is essentially solving the integration problem as in (9). We denote the integration problem as

$$I(f) = \int_\Gamma f(u)\rho(u)du,$$

where $u = (u_1, \ldots, u_d) \in \mathbb{R}^d$. Solving this integration problem often involves utilizing the numerical quadrature approach, which approximates the integral as a weighted sum of function evaluations at specific quadrature points:

$$I(f) \approx \sum_{i=1}^{n} w^{(i)} f(u^{(i)}), \tag{10}$$

where $u^{(i)} \in \Gamma$ and $w^{(i)} > 0$ represent the nodes and weights, respectively, to be determined by the quadrature rule.

The objective of the quadrature rule is to effectively approximate a large set of functions with a minimal number of function evaluations. This is typically achieved by ensuring equality conditions hold for all functions f within a polynomial subspace Π, as expressed by:

$$\int_\Gamma f(u)\rho(u)du = \sum_{i=1}^{n} w^{(i)} f(u^{(i)}) \quad \text{for all} \quad f \in \Pi_r. \tag{11}$$

Here, we consider the cross-polynomial subspace with a total polynomial order of r, defined as:

$$\Pi_r = \text{span}\{u^\alpha \mid \alpha \in \Lambda_r\}, \tag{12}$$

where

$$\alpha = (\alpha_1, \ldots, \alpha_d), \quad u^\alpha = \prod_{j=1}^{d} (u_j)^{\alpha_j} \tag{13}$$

and

$$\Lambda_r = \left\{ \alpha \mid \sum_{i=1}^{d} \alpha_i \leq r \right\}. \tag{14}$$

In many numerous integration problems, the evaluation functions exhibit smooth behavior and can be accurately approximated by the polynomials within the cross-polynomial subspace. In such instances, the quadrature rule enforcing Equation (11) can prove to be highly effective.

1. Gauss quadrature rule

One approach to generating the quadrature points is through the use of the Gauss quadrature rule [18]. In the one-dimensional case, the quadrature points $u^{(1)}, \ldots, u^{(k)}$ are the roots of the orthogonal polynomial $p_k(u)$ with degree k. The polynomials here are defined in the same way as the PCE basis functions, and they also satisfy orthogonality:

$$\left\langle p_i, p_j \right\rangle = \int_{\Gamma} p_i(u) p_j(u) \rho(u) du = \delta_{ij}. \tag{15}$$

The weights of the Gauss quadrature rule, $w^{(1)}, \ldots, w^{(k)}$ can be computed by solving the moment-matching equations:

$$\sum_{i=1}^{k} p_j \left(u^{(i)} \right) w^{(i)} = \begin{cases} 1 & \text{if } j = 0 \\ 0 & \text{for } j = 1, \ldots, k-1. \end{cases} \tag{16}$$

In this approach, the univariate Gauss quadrature rule with k nodes and weights can exactly integrate the univariate polynomials up to degree $2k - 1$.

For multi-dimensional integrations, the Gauss quadrature rule extends the univariate rule to form a tensor structure, Assuming k quadrature points in each dimension, the integral is approximated as

$$I \approx \sum_{i_1=0}^{k} \cdots \sum_{i_d=0}^{k} w_1^{(i_1)} \ldots w_d^{(i_d)} f(u_1^{(i_1)}, \ldots, u_d^{(i_d)}) \rho(u_1^{(i_1)}, \ldots, u_n^{(i_d)}), \tag{17}$$

where $(u_i^{(1)}, \ldots, u_i^{(k)})$ and $(w_i^{(1)}, \ldots, w_i^{(k)})$ are the nodes and weights for the 1D Gauss quadrature rule in the u_i dimension, respectively. We represent the full-grid quadrature points as

$$\boldsymbol{u} = \boldsymbol{u}_1^k \times \cdots \times \boldsymbol{u}_d^k, \tag{18}$$

where $\boldsymbol{u}_i^k := \{u_i^{(j)}\}_{j=1}^k$, and $\times$ represents the combination of quadrature points across dimensions using a tensor product. Indeed, while this method allows exact integration of cross-polynomials up to a total order of $2k - 1$, it suffers from the curse of dimensionality. This is because the total number of quadrature points increases exponentially with the number of dimensions, as $n = k^d$.

To mitigate the curse of dimensionality the Gauss quadrature rule suffers from, the Smolyak sparse grid method [19] offers a solution. This method strategically drops higher-order cross terms in the full-grid Gauss quadrature points, reducing the number of required quadrature points while preserving the accuracy of the quadrature rule to a significant extent. The sparse grid quadrature points can be expressed as:

$$\boldsymbol{u} = \bigcup_{\ell - d + 1 \leq |\mathbf{i}| \leq \ell} \left(\boldsymbol{u}_1^{i_1} \times \cdots \times \boldsymbol{u}_d^{i_n} \right), \tag{19}$$

where ℓ is the level of construction.

2. Designed quadrature method

For high-dimensional integration problems, a more effective quadrature rule is offered by the designed quadrature method proposed by Keshavarzzadeh et al. [20]. The core idea of the designed quadrature method is to numerically generate a quadrature rule that satisfies the multi-variate moment-matching equations:

$$\sum_{i=1}^{n} \Phi_{\boldsymbol{i}'} \left(u^{(i)} \right) w^{(i)} = \begin{cases} 1 & \text{if } |\boldsymbol{i}'| = 0 \\ 0 & \text{for } \alpha \in 0 < |\boldsymbol{i}'| < k, \end{cases} \tag{20}$$

where

$$\pi_\alpha = \prod_{i=1}^{d} p_{\alpha_i}^{(i)}(u_i) \tag{21}$$

and $p_j^{(i)}$ is the j-th orthogonal polynomial in the u_i dimension. If the nodes and weights of the quadrature rule satisfy the equations in (20), then the quadrature rules can exactly integrate cross-term polynomials up to $(2k-1)$th order. We denote the quadrature points and the weights as $\boldsymbol{u} := (u^{(1)}, \ldots, u^{(n)})$ and the weights $\boldsymbol{w} := (w^{(1)}, \ldots, w^{(n)})$ respectively. Then, the moment-matching equations in (20) can be written as residual equations:

$$\mathcal{R}(\boldsymbol{u}, \boldsymbol{w}) = 0, \tag{22}$$

where $\mathcal{R} : \mathbb{R}^{d\times n} \times \mathbb{R}^n \to \mathbb{R}^n$.

The designed quadrature method approximates the solution of the moment-matching equations by solving an optimization problem formulated as

$$\begin{aligned} &\min_{\boldsymbol{u},\boldsymbol{w}} \|\mathcal{R}(\boldsymbol{u}, \boldsymbol{w})\|_2 \\ &\text{subject to } u^{(j)} \in \Gamma, \quad j = 1, \ldots, n, \\ &\qquad\qquad w^{(j)} > 0, \quad j = 1, \ldots, n. \end{aligned} \tag{23}$$

The objective of this optimization, as shown in Equation (23), is to minimize the L^2 norm of the residual equations while constraining the nodes to lie within the support range and ensuring that the weights are positive. This method has demonstrated superior effectiveness compared to the sparse-grid Gauss quadrature rule for many high-dimensional integration problems. The number of quadrature points required is typically chosen as $n = \eta \frac{(2d)^{k-1}}{(k-1)!}$, where $\eta \in [0.5, 0.9]$ serves as a tuning parameter. The theoretical derivations for selecting the number of quadrature points, along with the strategies for optimization initialization, are thoroughly discussed in [20].

3. Other popular numerical quadrature methods

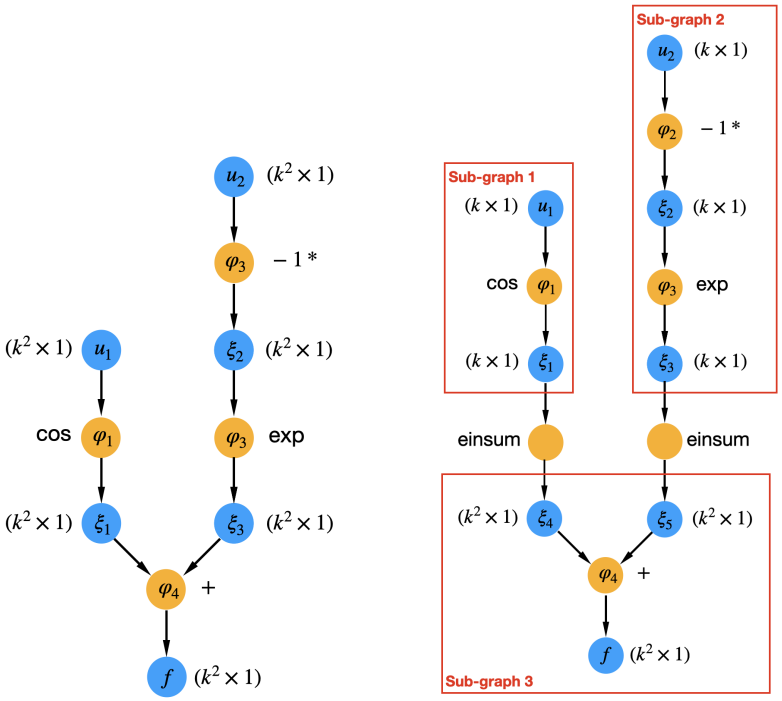
High-dimensional integration problems are prevalent in numerous fields beyond UQ. Over the years, many popular numerical quadrature methods have been developed to address these problems in various forms. In isogeometric analysis, B-spline quadrature rules have been found to be more efficient than Gauss quadrature rules [25–28], as they can approximate smooth functions using fewer basis functions. However, their application with PCE for solving UQ problems has only recently been explored [29]. Additionally, quasi-Monte Carlo methods [30, 31] represent another class of widely used numerical quadrature methods. Unlike Gauss quadrature, which assesses accuracy based on polynomial exactness of integration, quasi-Monte Carlo methods evaluate integration accuracy based on the discrepancy properties of the sample points.

C. Computational graph transformations for efficient tensor-grid evaluations

Accelerated Model evaluations on Tensor grids using Computational graph transformations (AMTC) is a computational graph transformation method, recently introduced by Wang et al. in [21], with its core idea initially proposed in [32]. AMTC accelerates the evaluation time of a computational model on tensor-grid inputs by eliminating repeated evaluations on the operation level. Any computational model can be represented as a computational graph with basic operations. In a conventional for-loop approach, when evaluating a computational model on tensor-grid inputs, each operation on the computational graph needs to be evaluated at every input point. However, with tensor-grid inputs, where inputs possess a fully tensorial structure, only a small set of unique values exists in each input space. Consequently, many operations within the computational model might depend on only a few inputs, leading to many redundant repeated evaluations at the operation level with a for-loop approach. To address this inefficiency, AMTC eliminates the redundant evaluations by generating a modified computational graph. The modified computational graph is partitioned into smaller sub-graphs based upon on which inputs each operation depends. Operations within each sub-graph share the same input space and are solely evaluated on distinct nodes within their input space. To ensure proper data flow between these sub-graphs, Einstein summation (*Einsum*) operations are inserted at each connection joint between sub-graphs.

To consider an illustrative example, assume we want to evaluate a simple function,

$$f = \cos(u_1) + \exp(-u_2), \tag{24}$$



(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 $f = cos(u_1) + exp(-u_2)$ [21]

at full-grid input points with k input points at each dimension. The given function can be represented as a sequence of elementary operations:

$$\begin{aligned} \xi_1 &= \varphi_1(u_1) = \cos(u_1); \\ \xi_2 &= \varphi_2(u_2) = -u_2; \\ \xi_3 &= \varphi_3(\xi_2) = \exp(\xi_2); \\ f &= \varphi_4(\xi_1, \xi_3) = \xi_1 + \xi_3. \end{aligned} \tag{25}$$

The tensor-grid input points are denoted as

$$\boldsymbol{u} = \begin{Bmatrix} (u_1^{(1)}, u_2^{(1)}) & \dots & (u_1^{(k)}, u_2^{(1)}) \\ \vdots & \ddots & \vdots \\ (u_1^{(1)}, u_2^{(k)}) & \dots & (u_1^{(k)}, u_2^{(k)}) \end{Bmatrix}. \tag{26}$$

The computational graphs to perform these full-grid evaluations with and without the AMTC method are shown in Fig. 1. In comparison, without AMTC, each operation in the computational graph is evaluated k^2 times, corresponding to the total number of input points. However, by applying AMTC, operations that depend only on u_1 and u_2 are only evaluated k times since their outputs have only k distinct values.

AMTC has been integrated into the middle end of the compiler for the *Computational System Design Language* (CSDL) [33], a recently developed domain-specific language tailored for tackling multidisciplinary design, analysis, and optimization problems. CSDL has gained significant popularity in addressing large-scale multidisciplinary design optimizations, with applications including designing novel air-taxi [34, 35] and spacecraft [36] concepts. Within the CSDL compiler, the AMTC serves as a middle-end algorithm, automatically generating the modified computational graph to minimize the model evaluation cost on tensor-grid inputs.

This method has been used in combination with the integration-based NIPC method, which forms full-grid quadrature rules for low-dimensional UQ problems. We refer to the combination of integration-based NIPC and

AMTC as *graph-accelerated NIPC*, and this method has demonstrated exceptional effectiveness in solving certain low-dimensional UQ problems involving multidisciplinary or multi-point systems [37].

III. Methodology

This paper proposes a new framework that generates a tensor-structured quadrature rule that is specifically tailored for a given computational model. This quadrature rule is intended to be used with the graph-accelerated NIPC method to efficiently solve the UQ problem.

A. Generation of tensor-structured quadrature rules

We use the degree of polynomials that a quadrature rule can exactly integrate as the measure of its accuracy. Under this criterion, if we maintain a non-tensorial structure for the quadrature rule, the most effective strategy is to utilize the designed quadrature method. In this scenario, the quadrature points can be expressed as:

$$\boldsymbol{u} = \boldsymbol{u}^{n}_{\{1,\dots,d\}}. \tag{27}$$

The nodes and weights in the quadrature rule are determined by solving the optimization problem outlined in (23), and the number of quadrature points is chosen as

$$n = \eta \frac{(2d)^{k-1}}{(k-1)!}, \quad \eta \in [0.5, 0.9]. \tag{28}$$

Adopting this method ensures that the resulting quadrature rule can effectively integrate polynomials up to $(2k-1)$th order. If a fully tensorial structure for the quadrature points is necessary, the optimal method to employ is the full-grid Gauss quadrature method. In this approach, the quadrature points are expressed as:

$$\boldsymbol{u} = \boldsymbol{u}^{k}_{\{1\}} \times \cdots \times \boldsymbol{u}^{k}_{\{d\}}, \tag{29}$$

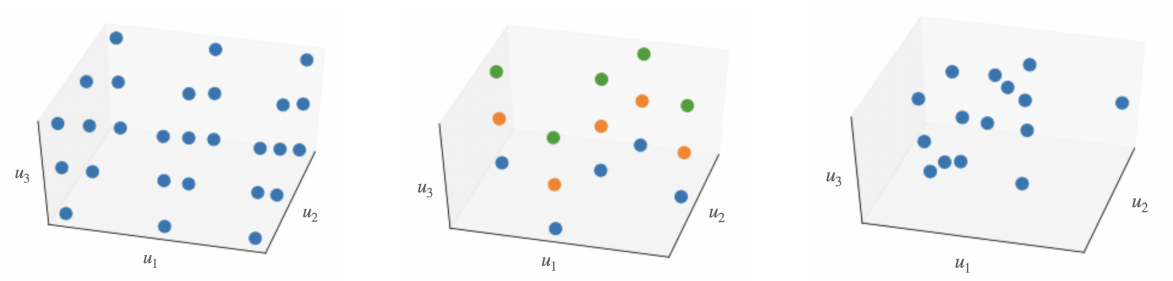
where $\boldsymbol{u}^{k}_{\{i\}}$ represents the 1D Gauss quadrature points in u_i dimension. The sets of 1D Gauss quadrature points in each dimension are determined separately to ensure the nodes and weights can exactly integrate the univariate polynomials up to $(2k-1)$th order. Maintaining a fully tensorial structure of the 1D Gauss quadrature points and weights ensures that the full-grid quadrature rule can exactly integrate cross-polynomials up to the $(2k-1)$th order, albeit while requiring $n = k^d$ quadrature points. In fact, in each dimension, the 1D Gauss quadrature points and weights correspond to the global minimum of the optimization problem outlined in (23). Assuming that the optimization problem in (23) always finds the global minimum, employing the full-grid Gauss quadrature rule is equivalent to applying the designed quadrature method in each dimension and subsequently forming a fully tensorial structure between the 1D solutions.

In addition to the fully tensorial and non-tensorial structure options, we can also preserve a partially tensorial structure for the quadrature points while ensuring accurate integration of all polynomials up to a specified order. For instance, in a 4D integration problem, we could maintain a partially tensorial structure as:

$$\boldsymbol{u} = \boldsymbol{u}^{n_1}_{\{1,2\}} \times \boldsymbol{u}^{n_2}_{\{3,4\}}, \tag{30}$$

where a tensorial structure is retained between the quadrature points in the space of u_1, u_2 and the quadrature points in the space of u_3, u_4. If the quadrature rules in each space are selected such that the quadrature points can precisely integrate polynomials up to the same order, then this partially tensorial quadrature rule can also precisely integrate the polynomials to the same order in the space of u. The total number of tensorial structure options for the quadrature rule can be represented by the Bell number from combinatorial mathematics, which is given by:

$$B_d = \sum_{i=0}^{d} \left\{ \begin{array}{c} i \\ d \end{array} \right\}. \tag{31}$$



(a) $\boldsymbol{u} = \boldsymbol{u}^n_{\{1\}} \times \boldsymbol{u}^n_{\{2\}} \times \boldsymbol{u}^n_{\{3\}}$ **(b)** $\boldsymbol{u} = \boldsymbol{u}^{n_1}_{\{1,2\}} \times \boldsymbol{u}^{n_2}_{\{3\}}$ **(c)** $\boldsymbol{u} = \boldsymbol{u}^n_{\{1,2,3\}}$

Fig. 2 Visualizations of quadrature points with different tensorial structure options

For example, for $d = 3$, $B_3 = 5$, the five options of the tensorial structure are:

$$\begin{aligned}
&\boldsymbol{u} = \boldsymbol{u}^n_{\{1,2,3\}},\\
&\boldsymbol{u} = \boldsymbol{u}^{n_1}_{\{1,2\}} \times \boldsymbol{u}^{n_2}_{\{3\}},\\
&\boldsymbol{u} = \boldsymbol{u}^{n_1}_{\{1,3\}} \times \boldsymbol{u}^{n_2}_{\{2\}},\\
&\boldsymbol{u} = \boldsymbol{u}^{n_1}_{\{2,3\}} \times \boldsymbol{u}^{n_2}_{\{1\}},\\
&\boldsymbol{u} = \boldsymbol{u}^{n_1}_{\{1\}} \times \boldsymbol{u}^{n_2}_{\{2\}} \times \boldsymbol{u}^{n_3}_{\{3\}}.
\end{aligned} \tag{32}$$

Visualizations of quadrature points with different tensorial structures can be found in Fig. 2.

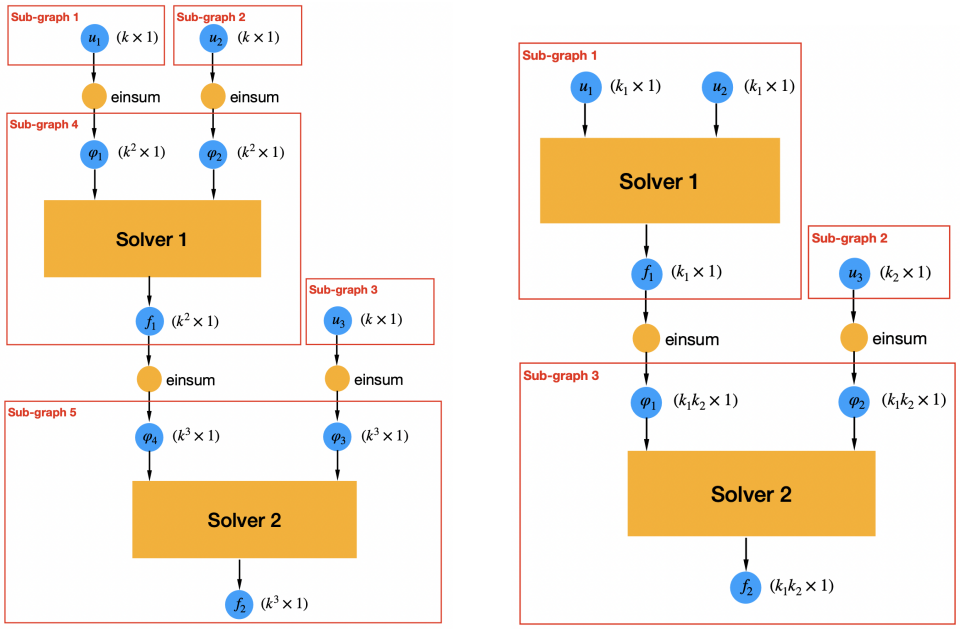
Given a specific tensorial structure option for the quadrature rule, we need to decide on the method for generating the quadrature points within each space and determine the requisite number of quadrature points to attain the desired level of accuracy. In our method, for a particular tensorial structure of the quadrature rule, we employ the designed quadrature method in each space. This method determines the nodes and weights to accurately integrate polynomials in that space up to a specified order. Adapted from the Gauss quadrature rule, we use the level of accuracy k for a quadrature rule to indicate its capability of precisely integrating polynomials up to the $(2k-1)$th order in the integration space. The number of quadrature points utilized in the designed quadrature method is given by (28), tailored for high-dimensional integration problems. In our setting, we also utilize the designed quadrature method for relatively low-dimensional spaces. Therefore, we choose the required number of quadrature points using:

$$n = \begin{cases} \frac{k^d}{d} & \text{for } d \leq 2 \\ 0.9\frac{(2d)^{k-1}}{(k-1)!} & \text{for } d > 2. \end{cases} \tag{33}$$

When the number of dimensions is greater than two, we follow the designed quadrature rule in (28) with $\eta = 0.9$. However, for 1D and 2D cases, we choose $n = \frac{k^d}{d}$. In this way, in the 1D space, it chooses k quadrature points to achieve k level of accuracy, which matches the Gauss quadrature rule. Consequently, for the 1D space, we can avoid solving the optimization problem in the designed quadrature method and directly use the 1D Gauss quadrature rule, as it represents the exact solution of the designed quadrature method. In 2D space, we have $n = \frac{k^2}{2}$, this has been numerically tested effective to generate sufficient quadrature points to achieve k level of accuracy for the quadrature rule. This approach results in generating the full-grid Gauss quadrature rule if a fully tensorial structure option is chosen, and generating the designed quadrature rule if a non-tensorial structure option is chosen.

B. Tensor-grid evaluations with AMTC

When not using the AMTC method for model evaluations, tensor-structured quadrature rules often underperform compared to non-tensorial quadrature rules. This is because tensor-structured quadrature rules always require evaluations of the model at more input points than non-tensorial quadrature rules to achieve the same level of accuracy. However, when utilizing the computational graph transformation method, AMTC, redundant operations incurred by the tensorial structure of the input points are eliminated at the operation level. Consequently, the overall cost of evaluating the model



(a) $\boldsymbol{u} = \boldsymbol{u}^k_{\{1\}} \times \boldsymbol{u}^k_{\{2\}} \times \boldsymbol{u}^k_{\{3\}}$ **(b)** $\boldsymbol{u} = \boldsymbol{u}^{k_1}_{\{1,2\}} \times \boldsymbol{u}^{k_2}_{\{3\}}$

Fig. 3 Computational graphs after AMTC on fully and partially tensor structured input points

is no longer directly tied to the number of quadrature points utilized. This gives us the reason for devising a custom tensor-structured quadrature rule tailored for a given computational model.

For instance, consider a 3D UQ problem involving two computationally expensive solvers. The computational model can be written as

$$\begin{aligned} f_1 &= \mathcal{F}_1(u_1, u_2), \\ f_2 &= \mathcal{F}_2(f_1, u_3), \end{aligned} \tag{34}$$

where $\mathcal{F}_1$ and $\mathcal{F}_2$ represent the first and second solver, respectively, and the UQ problem aims to compute the risk measures of f_2 under the effect of the three uncertain inputs. In this case, we have two uncertain inputs associated with solver 1, and one uncertain input associated with solver 2. Consequently, the operations in solver 1 are dependent on u_1 and u_2 while the operations in solver 2 are dependent on u_1, u_2, and u_3. In such scenario, an optimal quadrature rule can be easily formulated following a partially tensorial structure:

$$\boldsymbol{u} = \boldsymbol{u}^{k_1}_{\{1,2\}} \times \boldsymbol{u}^{k_2}_{\{3\}}. \tag{35}$$

The corresponding model evaluations can be significantly accelerated using the AMTC method. To demonstrate, we show the computational graphs when performing model evaluations on fully tensorial quadrature points $\boldsymbol{u}^k_{\{1\}} \times \boldsymbol{u}^k_{\{2\}} \times \boldsymbol{u}^k_{\{3\}}$ and partially tensorial quadrature points $\boldsymbol{u}^{k_1}_{\{1,2\}} \times \boldsymbol{u}^{k_2}_{\{3\}}$ after applying AMTC in Fig. 3. We observe that, in both cases, AMTC generates modified computational graphs that reduce the number of evaluations on solver 1. In comparison, the partially tensor-structured quadrature rule is more favorable than the fully tensor-structured quadrature rule, as both the u_1 and u_2 affect the same operations on the computational graph, so there is no need to maintain a tensor structure between u_1 and u_2 like in the fully tensor-structured quadrature rule. For the partially tensor-structured quadrature rule, by grouping the space of u_1 and u_2, the total number of quadrature points in this space, k_1, is smaller than the full-grid quadrature rule which has k^2 quadrature points in this space. Consequently, utilizing the partially tensor-structured quadrature rule results in fewer evaluations for both solver 1 and solver 2.

C. Finding a desirable tensor structure option

For specific UQ problems, particularly those involving multidisciplinary systems, certain partially tensor-structured quadrature rules demonstrate superior performance compared to existing ones when utilized with the graph-accelerated NIPC method. However, identifying an optimal tensor-structure option may be challenging.

Table 2 Operations dependency information for function $f = cos(u_1) + exp(-u_2)$

$D(\varphi_i, u_j)$	u_1	u_2
φ_1	1	0
φ_2	0	1
φ_3	0	1
φ_4	1	1

Theoretically, we can estimate the model evaluation costs after applying AMTC for each tensor-structured rule. To do that, for a computational model $f = \mathcal{F}(u)$, we denote the data and operations nodes as ξ_i and φ_j, respectively. Here l represents the number of operations in the computational graph. For example, consider the function

$$f = cos(u_1) + exp(-u_2), \tag{36}$$

with its corresponding computational graph described as

$$\begin{aligned} \xi_1 &= \varphi_1(u_1) = cos(u_1); \\ \xi_2 &= \varphi_2(u_2) = -u_2; \\ \xi_3 &= \varphi_3(\xi_1) = exp(\xi_2); \\ f &= \varphi_4(\xi_2, \xi_3) = \xi_2 + \xi_3. \end{aligned} \tag{37}$$

From the computational graph, we can generate the dependency information which stores whether the output of one operation is dependent on one input as a binary number, written as $D(\varphi_i, u_j)$. The dependency information can be viewed in a matrix. The dependency matrix for the function (36) is presented in Tab. 2, which indicates the dependencies between operations and input variables. To estimate the evaluation cost of each tensorial structure for the quadrature rule to achieve the k-level of accuracy, we also need to store the information regarding the evaluation cost of each operation on one quadrature point. This information can be represented either by the total number of floating-point operations (FLOPs) or the evaluation time. For the non-tensorial structure, if we assume all operations are dependent on at least one input, no repeated evaluations can be reduced by the AMTC method. Assume that the operations' cost on all of the quadrature points is roughly the same, the total model evaluation cost can be approximated as:

$$\text{Cost} \approx nO\left(f(u)\right) \approx n \sum_{i=1}^{l} O(\varphi_i), \tag{38}$$

where $O(\cdot)$ represents the evaluation cost of the function/operation and the number of quadrature points n is chosen as in (33). If we maintain a full-grid structure with k quadrature points in each dimension, the evaluation cost without the AMTC method is

$$\text{Cost} \approx k^d O\left(\mathcal{F}(u^{(1)})\right) \approx k^d \sum_{i=1}^{l} O(\varphi_i). \tag{39}$$

However, with the AMTC method, each operation in the graph is evaluated only for distinct values. In the full-grid case, if the output of an operation is dependent on $\tilde{d}$ number of uncertain inputs, there are only $k^{\tilde{d}}$ quadrature points in its dependent inputs space, and that operation will be evaluated for $k^{\tilde{d}}$ number of times using the AMTC method. In this case, we omit the cost of the Einstein summation nodes added to the computational graph by the AMTC method, the evaluation cost can be estimated as

$$\text{Cost} \approx \sum_{i=1}^{l} k^{\sum_{j=1}^{d} D(\varphi_i, u_j)} O(\varphi_i). \tag{40}$$

Using the same idea, we can also estimate the model evaluations cost for any tensorial structure option of the quadrature points. For example, consider the setting where we have a partially tensorial structure for the quadrature points, such as

$$\boldsymbol{u} = \boldsymbol{u}_{\{1,2\}}^{k_1} \times \boldsymbol{u}_{\{3\}}^{k_2}. \tag{41}$$

In our method, k_1 and k_2 are chosen from (33) using $d = 1$ and $d = 2$ respectively. In this case, there are k_1 quadrature points in the space of u_1 and u_2, and the k_2 quadrature points in the space of u_3. With the AMTC method, for the operations that are dependent on only u_1, only u_2 or only u_1 and u_2, they will be evaluated n_1 times. Operations dependent on only u_3, will be evaluated n_2 times. Operations not dependent on any input are evaluated once. The rest of the operations are evaluated $n_1 n_2$ times. The model evaluations cost for this tensor-structured quadrature rule can be estimated as

$$\text{Cost} \approx \sum_{i=1}^{l} k_1^{D(\varphi_i,u_1)D(\varphi_i,u_2)} k_2^{D(\varphi_i,u_3)} O(\varphi_i). \tag{42}$$

Estimating the total model evaluation cost for all tensor structure options enables us to choose the optimal one with the lowest estimated cost. However, the challenge lies in the fact that the total number of tensor structure options follows the Bell number as shown in (31), which scales significantly with the problem dimensions. To avoid this significant overhead cost associated with numerically finding the optimal tensor structure option, we propose an alternative approach. This approach involves selecting a desirable tensor structure option based on our understanding of the computational model and an analysis of the computational graph. By leveraging information from the computational graph and characteristics of the computational model, we can make informed decisions to choose a tensor structure that is likely to lead to efficient model evaluations. This approach balances computational efficiency with accuracy, making it a practical solution for many UQ problems with complicated computational models.

When dealing with computational models involving multiple disciplines or solvers, identifying a desirable tensor-structure option can sometimes be achieved by examining the computational graph at the solver level. For instance, in the example provided in (34), we can easily deduce the optimal tensor structure based on insights gained from the computational graph. In other cases, we can perform some analysis on the computational graph to help us decide on the desired tensor-structure option. Our rationale here is based on the observation that complex computational models typically contain one or two computationally expensive operations, such as linear or non-linear solvers, which significantly contribute to the overall computational cost. We aim to identify uncertain inputs that these expensive operations do not depend on and then establish a tensor structure to minimize the number of evaluations required for these operations, with the utilization of AMTC. To do that, we define the *sparsity ratio* (SR) of an uncertain input as the ratio of its dependent operations' cost over the total model evaluation cost which can be approximated as

$$\text{SR}(u_i) \approx \frac{\sum_{i=1}^{l} O(\varphi_i) D(\varphi_i, u_j)}{O(f(u))}, \tag{43}$$

where $\text{SR}(u_i)$ represents the sparsity ratio of u_i and it measures the percentage of the model evaluation cost that is dependent on u_i. We identify the sparse uncertain inputs based on the condition that their sparsity ratio is $< 5\%$. We then partition the uncertain inputs as

$$u = (u_s, u_{ns}), \tag{44}$$

where $u_s = (u_{s_1}, \ldots, u_{s_{d_1}}) \in \mathbb{R}^{d_1}$ is the set of sparse uncertain inputs and $u_{ns} = (u_{ns_1}, \ldots, u_{ns_{d_2}}) \in \mathbb{R}^{d_2}$ is the set of non-sparse uncertain inputs with $d = d_1 + d_2$.

In this case, if all of the uncertain inputs are sparse uncertain inputs, we may want to form a full-grid quadrature rule following the Gauss quadrature method, such that the quadrature points can be written as

$$\boldsymbol{u} = \boldsymbol{u}_{\{1\}}^{k} \times \ldots \times \boldsymbol{u}_{\{d\}}^{k}, \tag{45}$$

where $\boldsymbol{u}_{\{i\}}^{k}$ represents the k quadrature points in u_i space. In this way, even though the number of quadrature points increases exponentially as $n = k^d$, the model evaluation cost can be significantly accelerated using the AMTC method which only evaluates each operation on the distinct quadrature points in the space of the uncertain inputs on which it depends. However, in most cases, for most UQ problems, there often only exist a few sparse uncertain inputs. In this case, we suggest choosing the quadrature rule that maintains a tensor structure between the quadrature points in the sparse uncertain input space and quadrature points in the space of non-sparse uncertain inputs, such that the quadrature points can be written as

$$\boldsymbol{u} = \boldsymbol{u}_{\{ns\}}^{k_1} \times \boldsymbol{u}_{\{s\}}^{k_2}, \tag{46}$$

where $\boldsymbol{u}_{\{ns\}}^{k_1}$ represents quadrature points in u_{ns} space with k_1 points and $\boldsymbol{u}_{\{s\}}^{k_2}$ represent the quadrature points in u_s space with k_2 points. This results in a total of $k_1 k_2$ quadrature points in the quadrature rule. However, when using the

AMTC method to modify the computational graph, the computational cost on the modified computational graph would scale roughly linearly with k_1 as most of the expensive operations in the computational graph would only depend on u_{ns} and would only be evaluated n_1 times. An alternative option is to formulate the quadrature rule as

$$\boldsymbol{u} = \boldsymbol{u}^{k_1}_{\{ns\}} \times \boldsymbol{u}^{k}_{\{s_1\}} \times \ldots, \boldsymbol{u}^{k}_{\{s_{d_1}\}}, \tag{47}$$

which forms a tensor structure between the space of each sparse uncertain input. This option results in more quadrature points but it is easier to generate the quadrature rule and can be more effective to use if we know the sparse uncertain inputs are associated with different solvers.

We detail the specific steps of applying this method with graph-accelerated NIPC in Algorithm 1, and a corresponding flowchart is provided in Fig. 4.

Algorithm 1 Partially tensor-structured quadrature rule for graph-accelerated NIPC

1: Specify a computational graph for a numerical model and the uncertain inputs, U
2: Specify the level of accuracy of the quadrature rule, k
3: Choose a desired tensor structure option based on the analysis of the computational graph
4: Compute the required number of quadrature points in each space following (28)
5: Compute the quadrature points and weights in each space by solving (23)
6: Formulate the quadrature rule using the chosen tensorial structure
7: Perform the model evaluations on the tensor-structured quadrature points using AMTC
8: Post-process the evaluations using NIPC to solve the UQ problem

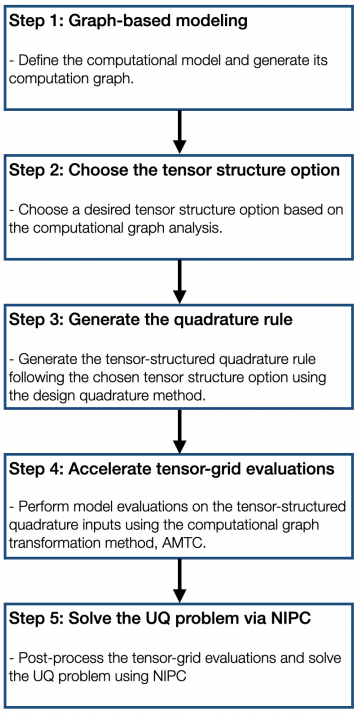


Fig. 4 A flow chart showing how the partially tensor-structured quadrature rule can be used with graph-accelerated NIPC to solve an UQ problem.

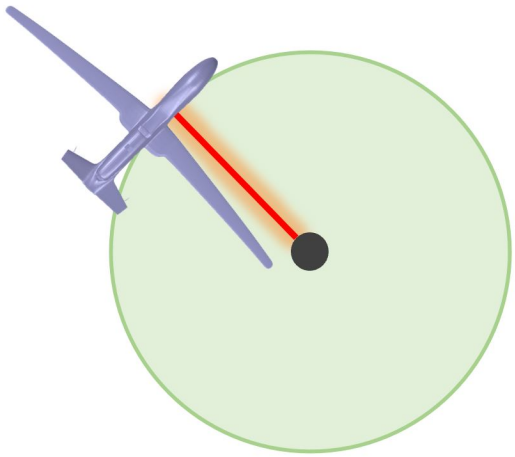


Fig. 5 A circular cruise mission around a ground station

Uncertain inputs	Distributions
Payload weight W_p (kg)	$\mathcal{N}(90, 10)$
Atmospheric extinction coefficient σ	$\mathcal{N}(0.2, 0.02)$
Flight altitude h (m)	$\mathcal{N}(10,000, 1000)$
Flight velocity v (m/s)	$\mathcal{N}(100, 10)$

Table 3 Uncertain inputs' distributions for the UAV problem

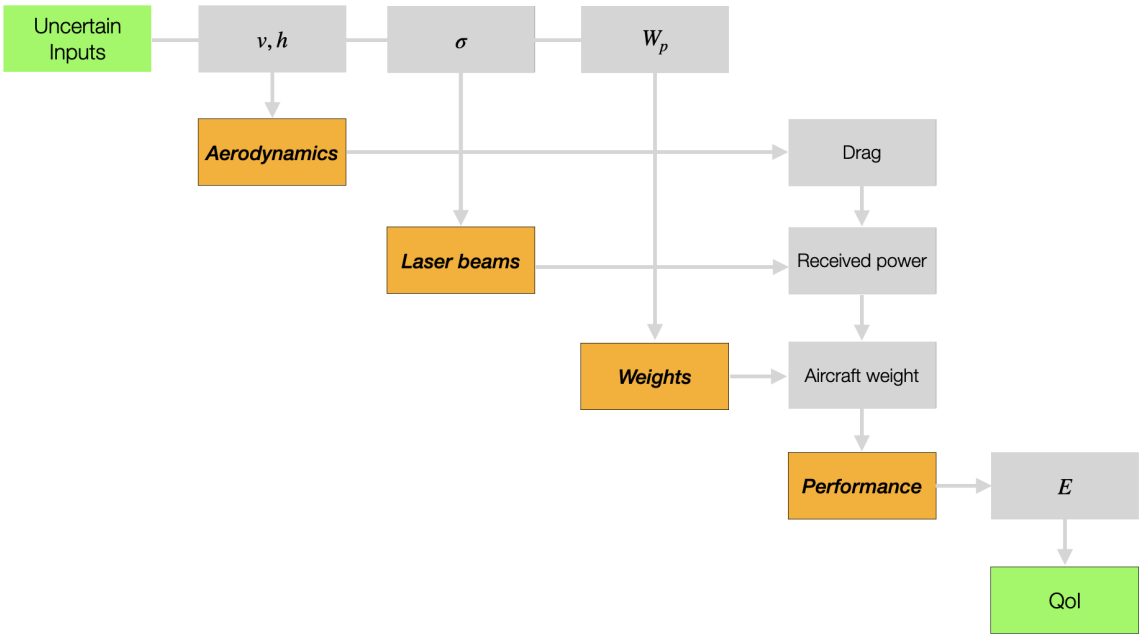


Fig. 6 Multidisciplinary structure of the UAV model

Table 4 Sparsity ratio of the uncertain inputs in the UAV model

Uncertain inputs	Sparsity ratio
v	92%
h	96%
W_p	4%
σ	2%

IV. Numerical Results

A. Laser-beam powered UAV multidisciplinary model

The first test problem is a 4D UQ problem adapted from a UAV design scenario [38]. This computational model assesses the total energy stored during a cruise mission undertaken by a laser-beam-powered UAV as it orbits a ground station for one complete cycle while being recharged by the laser beam emitted from the ground station. The mission is demonstrated in Fig. 5. We consider four uncertain inputs associated with the operating conditions, which are flight velocity, flight altitude, payload weight, and atmospheric extinction coefficient. Their distributions are shown in Tab. 3. The objective of this UQ problem is to compute the expectation of the stored energy considering the influence of these uncertain inputs.

The computational model used in this problem involves four disciplines: aerodynamics, laser beams, weights, and performance. The multidisciplinary structure of the computational model is shown in Fig. 6. For the laser-beam model, we apply the Beers–Lambert law for optical power transfer [39] to compute the power received by the aircraft as

$$P_{\text{rec}} = \eta P_{\text{tra}}, \quad \eta = e^{-\sigma R}, \tag{48}$$

where P_{rec} and P_{tra} denote the power received and transmitted, respectively; σ is the atmospheric extinction coefficient which differs across weather conditions; and R is the distance between the aircraft and the power source. In the aerodynamic model, inputs such as flight velocity, flight altitude, and the total aircraft weight are considered. This model is responsible for computing the total drag force exerted on the wing to sustain steady-state flight. Additionally, it determines the lift coefficient and drag coefficient by employing a linear model and a quadratic polar model, respectively, derived from airfoil data. The weights model estimates the weight of each aircraft component using the statistical equations from [40] and outputs the total aircraft weight. The performance model takes inputs such as flight velocity and total drag force to compute the total energy stored during the mission. This computation is based on the following equations:

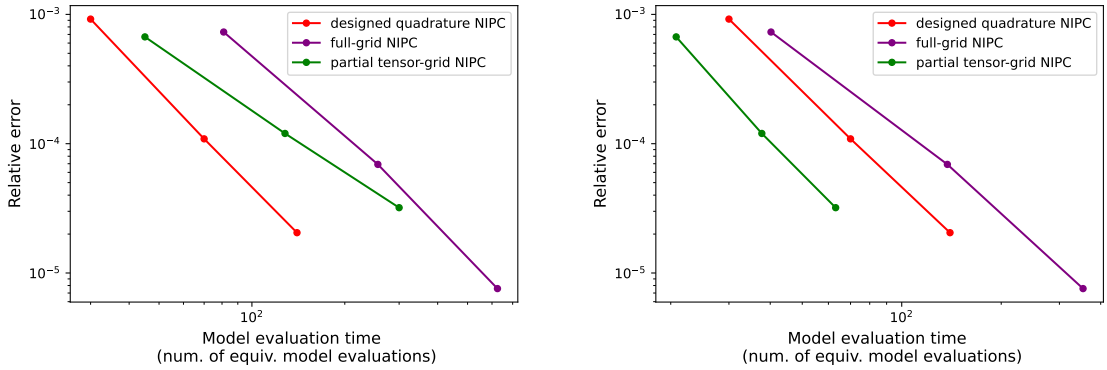
$$\begin{aligned} n &= \sqrt{\left(\frac{v}{rg}\right)^2 + 1}, \\ P_{\text{req}} &= \frac{D}{n}, \\ E &= (P_{\text{rec}} - P_{\text{req}})t, \end{aligned} \tag{49}$$

where n represents the turn load, r is the radius of the mission, g is the gravitational acceleration, D denotes the total drag force, P_{req} is the required power to maintain steady flight, and t represents the mission time.

On this test problem, following our proposed method, sparsity ratios of the uncertain inputs are measured, which are shown in Tab. 4. Using the sparsity ratios, two sparse uncertain inputs are identified: W_p and σ, as both of them have a sparsity ratio less than 5%. Combining this information with our understanding of the computational model, we constructed the partially tensor-structured quadrature rule as:

$$\boldsymbol{u} = \boldsymbol{u}^{n_1}_{\{h,v\}} \times \boldsymbol{u}^{n_2}_{\{W_p\}} \times \boldsymbol{u}^{n_3}_{\{\sigma\}}. \tag{50}$$

Our proposed method is compared with the designed quadrature method and the full-grid Gauss quadrature method, all of which are used with the NIPC method to solve the UQ problem. The UQ convergence plots, with and without using AMTC, are shown in Fig. 7. From the results, we observe that the partially tensor-structured quadrature rule is outperformed by the designed quadrature method when AMTC is not used to accelerate the model evaluations. This finding aligns with our theoretical analysis, as the non-tensorial quadrature rule requires fewer quadrature points



(a) Convergence plots without AMTC **(b) Convergence plost with AMTC**

Fig. 7 Convergence plots with and without AMTC for the UAV problem

than the tensor-structured quadrature rules to achieve the same level of accuracy. However, when we utilize AMTC to accelerate the model evaluations, the advantages of the structured quadrature rules become evident. Both the model evaluations associated with the full-grid quadrature rule and the partially tensor-structured quadrature rule are significantly accelerated by using the AMTC method. In this scenario, the partially tensor-structured quadrature rule achieves the best performance, with more than 50% reduction in evaluation costs than the other two methods. This is because, our method constructed a tailored tensorial structure that makes the most effective use of the inherent sparsity within the computational graph of this multidisciplinary system, resulting in the lowest model evaluation costs under the same level of accuracy.

B. Air-taxi trajectory design multidisciplinary model

The second test problem is a 6D UQ problem adapted from an air-taxi trajectory optimization scenario. This computational model calculates the average ground-level sound pressure level during the aircraft's flight along a specific trajectory. The UQ problem aims to determine the expected output under uncertainties in the control inputs and parameters of the acoustic model. Three uncertain inputs are associated with the initial conditions of the trajectory, and the other three uncertain inputs are associated with the parameters in the acoustic model. The distributions of the uncertain inputs can be found in Tab. 5. Notably, the control inputs used in this problem are generated from solving a deterministic trajectory optimization problem, which aims to minimize the total energy expended throughout the trajectory. Details of the computational model along with the trajectory optimization formulation can be found in [41].

Uncertain inputs	Distributions
Initial velocity v_0 (m/s)	$\mathcal{U}(30, 35)$
Initial flight path angel γ_0 (deg)	$\mathcal{U}(0, 2)$
Initial altitude h_0 (m)	$\mathcal{U}(0, 20)$
β_1	$\mathcal{N}(0.0209, 0.001)$
β_2	$\mathcal{N}(18.2429, 0.5)$
β_3	$\mathcal{N}(6.729, 0.2)$

Table 5 Uncertain inputs' distributions for the air-taxi problem

In this problem, the measured sparsity ratios are shown in Tab. 6. Based on this information, three uncertain inputs are identified as sparse uncertain inputs which are the three uncertain inputs associated with the acoustic model. This finding aligns with the multidisciplinary structure of the computational model, depicted in Fig. 8. From this figure, we observe that the computational model comprises four disciplines: flight dynamics, aerodynamics, propulsion, and acoustics, with two-way coupling among the first three disciplines. In this case, the coupled disciplines functions as a nonlinear operation and is solved iteratively. Due to its dominant role in computational costs, the three uncertain inputs associated with the nonlinear solver are the dense uncertain inputs in this computational model. Conversely, the

Table 6 Sparsity ratio of the uncertain inputs in the air-taxi model

Uncertain inputs	Sparsity ratio
v_0	99%
h_0	99%
γ_0	99%
β_1	2%
β_2	2%
β_3	2%

parameter uncertain inputs are the sparse uncertain inputs as they do not affect the upstream nonlinear solver in this computational model. Based on this reasoning, we choose the partially tensor-structured option as

$$\boldsymbol{u} = \boldsymbol{u}^{n_1}_{\{v_0,h_0,\gamma_0\}} \times \boldsymbol{u}^{n_2}_{\{\beta_1,\beta_2,\beta_3\}}, \tag{51}$$

in which we form a tensor structure between the space of dense uncertain inputs and the space of sparse uncertain inputs.

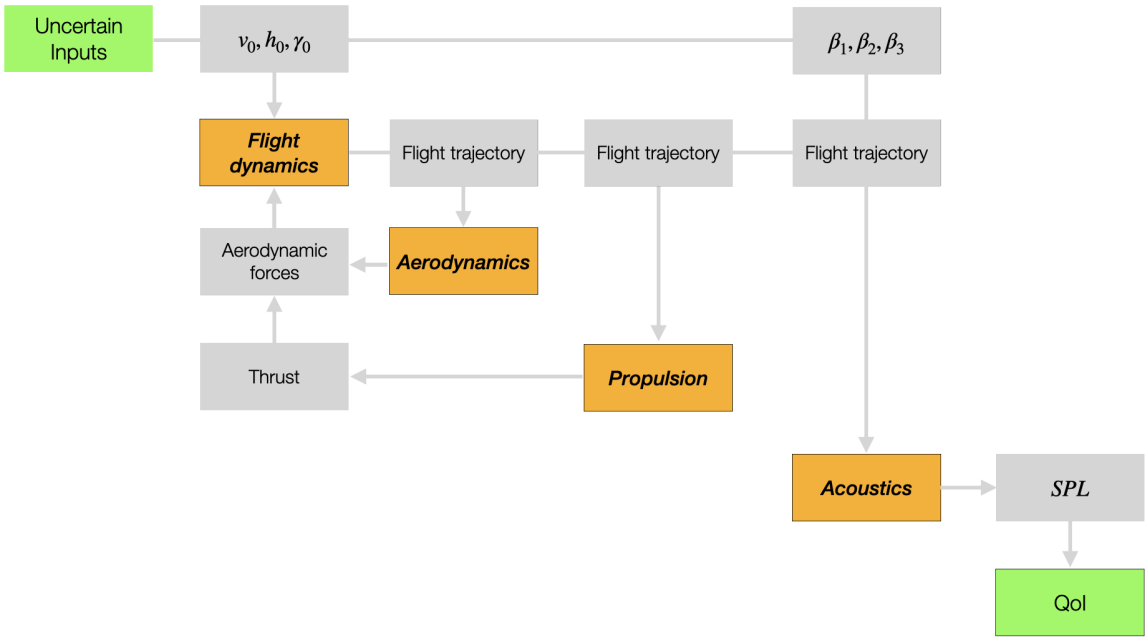
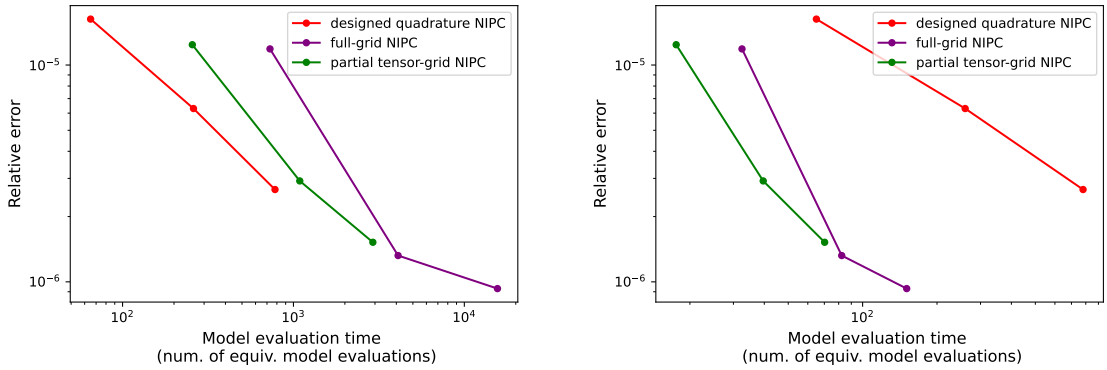


Fig. 8 Multidisciplinary structure of the air-taxi model

The UQ convergence plots with and without using AMTC are shown in Fig. 9. Similar to the first test problem, the partially tensor-structured quadrature rule is outperformed by the designed quadrature method when AMTC is not used. However, it becomes the most efficient method when using AMTC to accelerate the model evaluations. The reduction in computational costs is more than 40% in most cases. This can be explained by the fact that the tailored tensorial structure option takes full advantage of the inherent sparsity of the computational graph, resulting in the most efficient performance with the AMTC method. However, we also observe that when the desired relative error is exceptionally low (1e-6 and lower), the performance of the partially tensor-grid quadrature rule deteriorates on this test problem. This phenomenon can be attributed to the fact that the designed quadrature rule method can achieve similar performance to the Gauss quadrature rule while requiring a significantly smaller number of model evaluations. However, as the dimension of the optimization problem increases (as we have more optimization variables and more constraints), it becomes more challenging to solve the optimization problem to a tight tolerance using this method. Consequently, the



(a) Convergence plot without AMTC **(b) Convergence plot with AMTC**

Fig. 9 Convergence plots with and without AMTC for the airtaxi problem

designed quadrature method may struggle to achieve the same level of accuracy as the Gauss quadrature rule when an extremely high level of accuracy is required.

V. Conclusion

In this paper, we introduced a novel approach for generating partially tensor-structured quadrature rules tailored for utilization with the graph-accelerated NIPC method to tackle UQ problems. Our method entails initially choosing an appropriate tensorial structure option for the quadrature rule through an analysis of the computational graph. Subsequently, the quadrature rule possessing the chosen tensorial structure is generated using the designed quadrature method. This method has been tested on two UQ problems: one with four uncertain inputs and the other with six uncertain inputs, both involving multidisciplinary systems and are derived from aircraft design scenarios. The numerical results show that our method generates a new, partially tensor-structured quadrature rule for both problems. Notably, this new rule exhibits superior performance compared to both the existing designed quadrature method and the full-grid Gauss quadrature method when used with the graph-accelerated NIPC method.

One limitation of this proposed method is its reliance on the inherent sparsity of the computational graph, which restricts the generation of new quadrature rules for every problem. However, our method improves the efficiency of the graph-accelerated NIPC method, particularly for medium-dimensional UQ problems (typically with fewer than 10 dimensions) involving multidisciplinary systems. This is achieved by generating partially tensor-structured rules to more efficiently leverage the inherent sparsity within the computational graph. Another limitation lies in the potential performance decline of our method for high-dimensional UQ problems (ten dimensions or higher), where alternative approaches such as Monte Carlo methods may outperform it. A promising direction for future research involves integrating this method with dimension-reduction techniques to address higher-dimensional UQ problems. Another interesting research direction would be to develop non-tensor quadrature methods that are invariant under axis permutations, ensuring that flipping the axis vectors results in the same quadrature rule.

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] Abdar, M., Pourpanah, F., Hussain, S., Rezazadegan, D., Liu, L., Ghavamzadeh, M., Fieguth, P., Cao, X., Khosravi, A., Acharya, U. R., et al., "A review of uncertainty quantification in deep learning: Techniques, applications and challenges," *Information Fusion*, Vol. 76, 2021, pp. 243–297. https://doi.org/10.1016/j.inffus.2021.05.008.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] Xiu, D., and Karniadakis, G. E., "The Wiener–Askey polynomial chaos for stochastic differential equations," *SIAM journal on scientific computing*, Vol. 24, No. 2, 2002, pp. 619–644. https://doi.org/10.1137/S1064827501387826.

[11] 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.

[12] Hosder, S., Walters, R., and Balch, M., "Efficient sampling for non-intrusive polynomial chaos applications with multiple uncertain input variables," *48th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, 2007, p. 1939. https://doi.org/10.2514/6.2007-1939.

[13] Blatman, G., and Sudret, B., "Adaptive sparse polynomial chaos expansion based on least angle regression," *Journal of computational Physics*, Vol. 230, No. 6, 2011, pp. 2345–2367. https://doi.org/10.1016/j.jcp.2010.12.021.

[14] 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.

[15] Xiaojing, W., Zhang, W., Shufang, S., and Zhengyin, Y., "Sparse grid-based polynomial chaos expansion for aerodynamics of an airfoil with uncertainties," *Chinese Journal of Aeronautics*, Vol. 31, No. 5, 2018, pp. 997–1011. https://doi.org/10.1016/j.cja.2018.03.011.

[16] Luo, J., Xia, Z., and Liu, F., "Robust design optimization considering inlet flow angle variations of a turbine cascade," *Aerospace Science and Technology*, Vol. 116, 2021, p. 106893. https://doi.org/10.1016/j.ast.2021.106893.

[17] Eldred, M., and Burkardt, J., "Comparison of non-intrusive polynomial chaos and stochastic collocation methods for uncertainty quantification," *47th AIAA aerospace sciences meeting including the new horizons forum and aerospace exposition*, 2009, p. 976. https://doi.org/10.2514/6.2009-976.

[18] Golub, G. H., and Welsch, J. H., "Calculation of Gauss quadrature rules," *Mathematics of computation*, Vol. 23, No. 106, 1969, pp. 221–230. https://doi.org/10.1090/S0025-5718-69-99647-1.

[19] Smolyak, S. A., "Quadrature and interpolation formulas for tensor products of certain classes of functions," *Doklady Akademii Nauk*, Vol. 148, Russian Academy of Sciences, 1963, pp. 1042–1045.

[20] Keshavarzzadeh, V., Kirby, R. M., and Narayan, A., "Numerical integration in multiple dimensions with designed quadrature," *SIAM Journal on Scientific Computing*, Vol. 40, No. 4, 2018, pp. A2033–A2061. https://doi.org/10.1137/17M1137875.

[21] 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.

[22] Wiener, N., "The homogeneous chaos," *American Journal of Mathematics*, Vol. 60, No. 4, 1938, pp. 897–936. https://doi.org/10.2307/2371268.

[23] Ghanem, R., "Ingredients for a general purpose stochastic finite elements implementation," *Computer methods in applied mechanics and engineering*, Vol. 168, No. 1-4, 1999, pp. 19–34. https://doi.org/10.1016/S0045-7825(98)00106-6.

[24] Lee, D., and Rahman, S., "Practical uncertainty quantification analysis involving statistically dependent random variables," *Applied Mathematical Modelling*, Vol. 84, 2020, pp. 324–356. https://doi.org/10.1016/j.apm.2020.03.041.

[25] Johannessen, K. A., "Optimal quadrature for univariate and tensor product splines," *Computer Methods in Applied Mechanics and Engineering*, Vol. 316, 2017, pp. 84–99. https://doi.org/10.1016/j.cma.2016.04.030.

[26] Bartoň, M., and Calo, V. M., "Gauss–Galerkin quadrature rules for quadratic and cubic spline spaces and their application to isogeometric analysis," *Computer-Aided Design*, Vol. 82, 2017, pp. 57–67. https://doi.org/10.1016/j.cad.2016.07.003.

[27] Calabro, F., Sangalli, G., and Tani, M., "Fast formation of isogeometric Galerkin matrices by weighted quadrature," *Computer Methods in Applied Mechanics and Engineering*, Vol. 316, 2017, pp. 606–622. https://doi.org/10.1016/j.cma.2016.09.013.

[28] Bartoň, M., Puzyrev, V., Deng, Q., and Calo, V., "Efficient mass and stiffness matrix assembly via weighted Gaussian quadrature rules for B-splines," *Journal of Computational and Applied Mathematics*, Vol. 371, 2020, p. 112626. https://doi.org/10.1016/j.cam.2019.112626.

[29] Rahman, S., "A spline chaos expansion," *SIAM/ASA Journal on Uncertainty Quantification*, Vol. 8, No. 1, 2020, pp. 27–57. https://doi.org/10.1137/19M1239702.

[30] Niederreiter, H., *Random number generation and quasi-Monte Carlo methods*, SIAM, 1992.

[31] Sloan, I. H., and Woźniakowski, H., "When are quasi-Monte Carlo algorithms efficient for high dimensional integrals?" *Journal of Complexity*, Vol. 14, No. 1, 1998, pp. 1–33. https://doi.org/10.1006/jcom.1997.0463.

[32] 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.

[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] 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.

[35] Ruh, M. L., Fletcher, A., Sarojini, D., Sperry, M., Yan, J., Scotzniovsky, L., van Schie, S. P., Warner, M., Orndorff, N. C., Xiang, R., et al., "Large-scale multidisciplinary design optimization of a NASA air taxi concept using a comprehensive physics-based system model," *AIAA SCITECH 2024 Forum*, 2024, p. 0771. https://doi.org/10.2514/6.2024-0771.

[36] Gandarillas, V., and Hwang, J. T., "TALOS: A toolbox for spacecraft conceptual design," *arXiv preprint arXiv:2303.14936*, 2023. https://doi.org/10.48550/arXiv.2303.14936.

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

[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.

[39] Kim, I. I., McArthur, B., and Korevaar, E. J., "Comparison of laser beam propagation at 785 nm and 1550 nm in fog and haze for optical wireless communications," *Optical wireless communications III*, Vol. 4214, Spie, 2001, pp. 26–37. https://doi.org/10.1117/12.417512.

[40] Raymer, D., *Aircraft design: a conceptual approach*, American Institute of Aeronautics and Astronautics, Inc., 2012. https://doi.org/10.2514/4.869112.

[41] Orndorff, N. C., Sarojini, D., Scotzniovsky, L., Gill, H., Lee, S., Cheng, Z., Zhao, S., Mi, C., and Hwang, J. T., "Air-taxi transition trajectory optimization with physics-based models," *AIAA SCITECH 2023 Forum*, 2023, p. 0324. https://doi.org/10.2514/6.2023-0324.