High-dimensional uncertainty quantification using graph-accelerated non-intrusive polynomial chaos and active subspace methods

Bingran Wang*, Nicholas C. Orndorff†, Mark Sperry‡, and John T. Hwang§
University of California, San Diego, La Jolla, CA, 92093

Integration-based non-intrusive polynomial chaos (NIPC) is a widely used method for solving low-dimensional uncertainty quantification (UQ) problems that often forms the quadrature points in a tensor structure. A recently developed graph transformation method, Accelerated Model evaluations on Tensor grids using Computational graph transformations (AMTC) can significantly accelerate model evaluations on tensor-structured inputs when the computational graph is sparse. Integration-based NIPC with AMTC has proven effective for low-dimensional UQ problems, but it is limited by the curse of dimensionality in high-dimensional cases. To address this, we propose a new framework, AS-NIPC-AMTC, that connects this approach with the active subspace (AS) method, a popular dimension reduction technique. In the process of developing this new method, we have also developed a general framework, AS-NIPC that connects the integration-based NIPC with the AS method to solve high-dimensional UQ problems. This framework includes rigorous approaches to generate the orthogonal polynomial basis functions of the lower dimensional active subspace inputs and efficient quadrature rules to estimate their coefficients. The AS-NIPC-AMTC method extends the AS-NIPC method and generates the quadrature rule following a desired tensor structure so that the model evaluations can be significantly accelerated when AMTC is available. We demonstrate the effectiveness of both AS-NIPC and AS-NIPC-AMTC methods on an 81-dimensional UQ problem computing the average ground-level sound pressure of an electric air taxi flying a prescribed trajectory. Both methods outperform existing UQ methods when the number of function evaluations is limited to hundreds or fewer. The AS-NIPC-AMTC method results in at least 80% less error with a fixed function evaluation cost, while AS-NIPC results in 30% less error. The proposed methods provide a novel and rigorous way to combine the traditional NIPC method with the recently developed dimension reduction and graph transformation techniques, offering unique advantages when solving high-dimensional UQ and optimization under uncertainty problems.

I. Introduction

Uncertainty is ubiquitously present in many scientific applications and can significantly impact a system's behavior or performance. To address this, forward uncertainty quantification (UQ) (also known as uncertainty propagation) aims to quantify and analyze the impact of uncertain inputs on a system's output or quantity of interest (QoI) to provide reliable estimates of its uncertainty. These uncertainty estimates can then support decision-making and risk assessment. The applications of UQ can be found in many engineering disciplines such as fluids [1, 2], structures [3, 4], control [5] and machine learning [6].

Uncertainties in the inputs can arise from various sources and are generally categorized as aleatoric or epistemic. Aleatoric uncertainties, also known as irreducible uncertainties, are inherent to the system and may include variations in operating conditions, mission requirements, and model parameters, among others. These uncertainties can be defined within a probabilistic framework. Epistemic uncertainties, on the other hand, are reducible uncertainties that arise from a lack of knowledge. Such uncertainties may be caused by approximations used in the computational model or numerical solution methods and are difficult to define within a probabilistic framework.

*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.

This paper focuses on UQ within a probabilistic formalism where uncertain inputs are continuous random variables with known probability density distributions. The goal is to estimate statistical moments such as mean and variance of the QoI, or more complex metrics such as the probability of failure, conditional value at risk (CVaR) [7], or the probability density distribution of the QoI. Quantifying the impact of uncertainties on a system's output is critical in assessing the reliability of the system and enabling informed decision-making.

In the context of design optimization, UQ is a critical sub-problem of optimization under uncertainty (OUU) problems [8]. OUU aims to find an optimal solution that is robust to uncertainties. Uncertainty propagation plays a crucial role in this process as it determines the effect of input parameter uncertainties on the objective function or output variables, allowing the optimization algorithm to find a more robust and reliable solution. Without accurate uncertainty propagation, the optimization algorithm may converge to a solution that is not robust to uncertainties.

OUU problems can be broadly divided into two categories: robust design optimization (RDO) [9, 10] and reliability-based design optimization (RBDO) [11]. In RDO, the effects of uncertainty in the objective and constraints are considered through statistical moments, while in RBDO, probabilistic constraints are added based on the probability of failure. To solve OUU problems, the computational framework typically involves two nested for-loops of a UQ solver inside an optimizer. At each optimization iteration, an uncertainty propagation problem is solved to calculate the objective and constraint function, along with its derivatives if a gradient-based optimizer is used. In this case, the scalability and accuracy of the UQ method are critical factors in solving large-scale OUU problems.

UQ has been a major research area for many years and non-intrusive approaches have gained popularity due to their ease of implementation. The most common non-intrusive methods include the method of moments, polynomial chaos, kriging, and Monte Carlo. These methods have their strengths and weaknesses, and the choice of method depends on the problem at hand. The method of moments analytically computes the statistical moments of the QoI from the Taylor series approximation using the function evaluations and derivative information at one or a few fixed points [12]. This method provides an efficient way to estimate the statistical moments but is unable to compute more complicated risk measures of the QoI and suffers from the accuracy problem when the variance of inputs is significant. The Monte Carlo approach randomly samples the uncertain inputs and computes the risk measure of QoI using the corresponding output data. Due to the rule of large numbers, the convergence of Monte Carlo is independent of the number of uncertain inputs, which makes it the desirable method to use for solving high-dimensional problems. Recent research to improve its efficiency focuses on using multi-fidelity [13, 14] and importance sampling [15] approaches. However, for low-dimensional problems, the Monte Carlo method could still require significantly more model evaluations to achieve the same accuracy as other UQ methods. Kriging, also known as Gaussian process regression, constructs a response surface function using the input-output data. The trained kriging model acts as a surrogate model of the original function to allow a large number of model evaluations to conduct reliability analysis [16, 17] or optimization under uncertainty [18]. The polynomial chaos method represents the QoI as orthogonal polynomials that are derived from the distributions of the uncertain inputs. This method exploits the smoothness in the random space and enables a fast convergence rate with sampling or integration techniques. Common polynomial chaos-based methods include non-intrusive polynomial chaos [19–21] and stochastic collocation [22, 23]. Kriging and polynomial chaos are popular UQ approaches to use for low or medium-dimensional problems, However, they suffer from the curse of dimensionality and are not efficient enough to be directly applied to high-dimensional UQ problems.

Recently, a new method called Accelerated Model evaluations on Tensor grids using Computational graph transformations (AMTC) [24, 25] was proposed by Wang and Hwang to accelerate model evaluations on tensor-grid inputs. This method modifies the computational graph of the model from the middle-end of a modeling language to eliminate the repeated evaluations on the operational level which are incurred by the tensor structure of the inputs. This method has been well connected to the integration-based NIPC to generate quadrature rules with optimal tensor structures and can outperform the existing UQ methods when solving certain low-dimensional UQ problems [26]. However, this method still suffers the curse of dimensionality for many high-dimensional UQ problems.

For high-dimensionality UQ problems, one approach to reduce the computational cost is to apply dimension reduction techniques. Popular dimension reduction methods in uncertainty propagation problems include Sensitivity Analysis [27], Principal Component Analysis (PCA) [28], Partial Least Squares (PLS) [29], and Active Subspace (AS) [30]. Sensitivity Analysis is a commonly used method that determines the impact of input parameters on output uncertainty. Local sensitivity analysis measures the perturbation of each input at a nominal value and its effect on output, making it inexpensive to implement. However, this method may lack accuracy in many cases. In contrast, global sensitivity analysis [31] computes Sobol indices, which measure output variations over the full range of inputs, providing more accurate results but at a higher computational cost. PCA performs an eigenvalue decomposition on the covariance matrix of the uncertain inputs to identify directions of maximum variance in the input space. However, it is not suitable for independent uncertain inputs. PLS uses the regression method to identify directions that maximize covariance between input parameters and output uncertainty, but it may suffer from overfitting and does not utilize gradient information. On the other hand, AS uses gradient information to identify an active subspace where most of the first-order changes in the output exist. AS strikes a balance between accuracy and computational cost, making it a powerful tool for high-dimensional UQ problems, while also providing an intuitive way to visualize and interpret analysis results.

This paper aims to extend the capabilities of the graph-accelerated NIPC method (integration-based NIPC + AMTC) to solve high-dimensional UQ problems by leveraging the active subspace method. While the AMTC method has proven effective in solving low-dimensional UQ problems with sparse computational graph structures in conjunction with integration-based NIPC, there is currently no existing framework that can directly connect the graph-accelerated NIPC to the AS method. Although previous works [32, 33] have connected the AS method to regression-based NIPC methods, they cannot be used to connect the active subspace method to the graph-accelerated NIPC method, which relies on an integration-based approach to solve the polynomial chaos expansion (PCE) coefficients. To address this gap, this paper first proposes a general framework called AS-NIPC, which generates PCE basis functions for the active subspace's uncertain inputs and an efficient quadrature rule in the original uncertain input space to connect the active subspace method to the integration-based NIPC method. This framework can be applied to any high-dimensional UQ problem without the need for the AMTC method. Furthermore, an extension of this framework, AS-NIPC-AMTC, is presented to use with the AMTC method. This extension involves identifying sparse uncertain inputs using computational graph information, applying the AS method to non-sparse uncertain inputs, and generating a desired tensor-structured quadrature rule to take advantage of the sparse computational graph when the AMTC method is available.

The proposed methods have been tested on an 81-dimensional UQ problem computing the ground-level noise of a lift-plus-cruise electric air taxi flying a prescribed trajectory with parameter uncertainties and control input uncertainties. The results show that the proposed AS-NIPC and AS-NIPC-AMTC methods are more effective to use than the existing methods in the case we can afford hundreds or fewer evaluations of the model.

This paper is organized as follows. Section II gives some background for NIPC, AMTC, and Active subspace methods. Section III presents the details of the AS-NIPC and AS-NIPC-AMTC methods. Section IV shows the numerical results of the test problem. Section V summarizes the work and offers concluding thoughts.

II. Background

A. Non-intrusive polynomial chaos

Polynomial chaos, also known as polynomial chaos expansion (PCE), was first introduced by Wiener, who used Hermite polynomials to develop a physical theory for Gaussian random processes [34]. Ghanem extended this idea to use Hermite polynomials as an orthogonal basis to represent random processes [35]. However, PCE suffers from convergence issues for problems with non-Gaussian uncertain inputs. Xiu and Karniadakis proposed the generalized polynomial chaos (gPC) method to address this difficulty [36]. The gPC method utilizes different orthogonal polynomials for uncertain inputs with different distributions to achieve optimal convergence for the UQ problem.

Consider an uncertainty propagation problem with a multivariate stochastic function written as

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

where $U = (U_1, \ldots, U_d) \in \mathbb{R}^d$ are a set of mutually uncertain inputs with probability density distributions $\rho(u)$ and support Γ, $F \in \mathbb{R}$ is the QoI of this problem and $\mathcal{F} : \mathbb{R}^d \to \mathbb{R}$ is the evaluation function. The gPC method represents the QoI as an infinite series of orthogonal polynomials of the uncertain inputs:

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

where $\Phi_i(U)$ and α_i are PCE basis functions and their corresponding PCE coefficients, respectively. In practice, (2) is often truncated to a specific order, leading to an approximation function of the QoI with limited terms, written as

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

If we truncate the series by choosing only the polynomials up to the order of p, the resulting number of PCE terms follows

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

These PCE basis functions satisfy the orthogonality property as

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

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

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

Constructing the PCE basis functions is extremely important to ensure the fast convergence rate of any PCE-based methods, and this can be nontrivial in many cases. For one-dimensional problems, the univariate orthogonal polynomials have been derived for common types of continuous random variables, shown in Tab. 1. For multi-dimensional problems, if the inputs are mutually independent, the PCE basis functions can be formed as a tensor-product of the univariate orthogonal polynomials that correspond to each uncertain input. We here introduce the multi-index notation and define a p-tuple as

$$i' = (i_1, \ldots, i_p) \in \mathbb{N}_0^p, \tag{7}$$

with its magnitude defined as

$$|i'| = i_1 + \ldots + i_p. \tag{8}$$

With this notation, we let $\{\phi_i(\tilde{u}_k)\}_{i=0}^p$ denote the set of univariate PCE basis functions for variable u_k. Then the multivariate PCE basis with total degree up to p can be represented as

$$\Phi_{\boldsymbol{i}'}(\tilde{u}) = \phi_{i_1}(\tilde{u}_1)\ldots\phi_{i_d}(\tilde{u}_d), \quad |i'| \leq p. \tag{9}$$

However, this process becomes thorny if the uncertain inputs are not independent or they do not follow common types of distributions. For a special case, if the uncertain inputs are multivariate Gaussian random variables, the PCE basis functions can be analytically derived following [37]. For arbitrary and dependent uncertain inputs, the PCE basis functions can be numerically generated using a whitening transformation matrix as in [38].

There are mainly two approaches to solve the PCE coefficients α_i in (3): by integration or regression. These two approaches show similar performance with significantly higher convergence rates than the Monte Carlo method on low-dimensional UQ problems[39], but their research focuses are drastically different. The integration approach makes use of the orthogonality property of PCE basis functions in (5) and projects the QoI onto each basis function, which results into an integration problem for each coefficient as:

$$\begin{aligned} \alpha_i &= \frac{\left\langle F(U), \Phi_i \right\rangle}{\left\langle \Phi_i^2 \right\rangle} \\ &= \int_\Gamma f(u)\Phi_i(u)\rho(u)du \\ &= \int_{\Gamma_1} \ldots \int_{\Gamma_d} f(u_1, \ldots, u_d)\Phi_i(u_1, \ldots, u_d)\rho(u_1, \ldots, u_d)du_1 \ldots du_d. \end{aligned} \tag{10}$$

In this approach, the number of the PCE basis functions we use barely affects the computational cost as we use the same quadrature rule to estimate all of the PCE coefficients. The main difficulty here is to derive an efficient quadrature rule to solve the high-dimensional integration problem in (10). The easiest quadrature rule to use is the Gauss quadrature rule [40]. In 1D case, the Gauss quadrature rule is optimal to use as the quadrature rule with k nodes can exactly integrate polynomials up to $(2k-1)$th order. However, in the multi-dimensional case, the quadrature rule is formed as a tensor product of the 1D quadrature rule and the number of quadrature points increases exponentially with the dimension as k^d. More efficient quadrature rules to use for high-dimensional problems include Smolyak sparse-grid quadrature rule [41] which drops higher order terms in Gauss quadrature rule while maintaining minimal loss and the designed quadrature rule [42], which numerically generate the quadrature rule that satisfies the moment matching equations through optimization.

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	$[0, \infty)$

On the other hand, the regression approach computes the PCE coefficients as the least-square solution of the linear system

$$\begin{bmatrix} \Phi_1(u^{(1)}) & \dots & \Phi_q(u^{(1)}) \\ \vdots & & \vdots \\ \Phi_1(u^{(n)}) & \dots & \Phi_q(u^{(n)}) \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_q \end{bmatrix} = \begin{bmatrix} f(u^{(1)}) \\ \vdots \\ f(u^{(n)}) \end{bmatrix}, \tag{11}$$

where $(u^{(1)}, \dots, u^{(n)})$ represents a set of n sample points generated from the distributions of the uncertain inputs. The number of samples n is typically chosen to be 2-3 times the number of coefficients. This means the number of PCE basis functions used in NIPC would directly affect the required number of model evaluations and the accuracy of the UQ methods. Thus the main focuses to improve its efficiency are on sparse PCE basis selection [43, 44] and optimal sampling [45].

B. Accelerated model evaluations on tensor grids using computational graph transformation

The Accelerated Model evaluations on Tensor grids using Computational graph transformations(AMTC) method is a graph transformation method to reduce the model evaluation cost on tensor-grid input points. The AMTC method takes advantage of the fact that when we view a computational model as a computational graph with elementary operations, each operation's output only has distinct evaluations on the distinct nodes in the space of its dependent input. When evaluating a model on tensor-grid input nodes, often many operations do not depend on all of the uncertain inputs and the traditional framework to run the entire model in a for-loop approach creates many repeated evaluations on the operational level. The AMTC method eliminates repeated evaluations on the operational level by partitioning the computational graph into several sub-graphs 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. The Einstein summation nodes are inserted connecting the sub-graphs to ensure correct data flow between the sub-graphs. For example, suppose we want to evaluate a simple function

$$f = cos(u_1) + exp(-u_2) \tag{12}$$

on tensor-grid input points with k points in each dimension. The function can be decomposed into a sequence of elementary operations as

$$\begin{aligned} \xi_1 &= \varphi_1(u_1) = cos(u_1); \\ \xi_2 &= \varphi_2(u_2) = -u_2; \\ \xi_3 &= \varphi_3(\xi_2) = exp(\xi_2); \\ f &= \varphi_4(\xi_2, \xi_3) = \xi_2 + \xi_3, \end{aligned} \tag{13}$$

where φ_i represents the elementary operations in the function and the computational graph of this function is shown in Fig. 1. The input points are given as

$$\boldsymbol{u} = \begin{Bmatrix} (u_1^{(1)}, u_2^{(1)}) & \dots & (u_1^{(k)}, u_2^{(1)}) \\ \vdots & \ddots & \vdots \\ (u_1^{(1)}, u_2^{(k)}) & \dots & (u_1^{(k)}, u_2^{(k)}) \end{Bmatrix}, \tag{14}$$

with k^2 input points. The computational graphs to evaluate this function using and without using the AMTC method are shown in Fig. 2. When not using the AMTC method, all of the operations in the computational graph are evaluated k^2

times. However, by using the AMTC method, the operations that are only dependent on u_1 or u_2 are evaluated k times as their outputs only have k distinct values. The AMTC method has been well connected to the integration-based NIPC to solve the integration problems using the quadrature rule with full-grid quadrature points and also offers flexibility for the users to design the quadrature points with an optimal tensor structure based on the computational graph information [26].

This method has been implemented in the middle end of the compiler for a recently developed modeling language, Computational System Design Language (CSDL) [46]. CSDL is a domain-specific language designed for large-scale multi-disciplinary design analysis and optimization and generates the computational graph of the model based on the source code. When evaluating the model on tensor grids, the AMTC method is used as a middle-end algorithm that modifies the computational graph so that the elimination of repeated evaluations on the operational level can be achieved. A demonstration of the implementation of AMTC on CSDL is shown in Fig. 3.

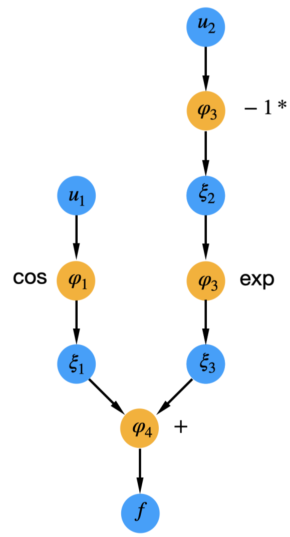


Fig. 1 Computational graph of function $f = cos(u_1) + exp(-u_2)$

C. Active subspace

The active subspace(AS) method traces back to Russi's Ph.D. thesis in [47]. The widely known form of it was proposed by Constantine et. al. in [30]. It is a popular dimension-reduction method for solving high-dimensional UQ problems and has also been used to solve optimization [48, 49] and inverse problems [50]. The basic idea behind AS is to exploit the fact that some directions in the input parameter space have a greater impact on the output uncertainty than others. By identifying these important directions, we can project the high-dimensional input space onto a lower-dimensional subspace that captures the most important input-output relationships. To achieve that, the AS method computes the mean squared gradients of the objective function with respect to the uncertain inputs as

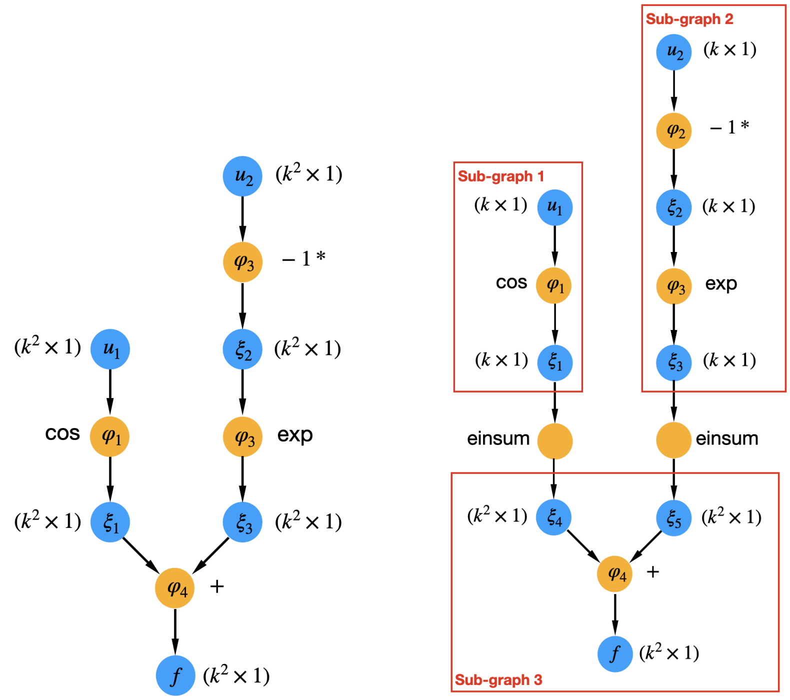
$$C = \mathbb{E}[(\nabla_u f)(\nabla_u f)^T] = \int_\Gamma (\nabla_u f)(\nabla_u f)^T \rho(u) du, \tag{15}$$

where $\nabla_u f = [\frac{\partial f}{\partial u_1}, \ldots, \frac{\partial f}{\partial u_d}]^T$. The C matrix is symmetric and positive semi-definite and thus has non-negative eigenvalues and orthogonal eigenvectors. Following an eigenvalue decomposition, the C matrix can be expressed as

$$C = W \Lambda W^T, \quad \Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_d). \tag{16}$$

The eigenvectors W define a rotation of the original uncertain input space and are sorted by the magnitude of their corresponding eigenvalues in decreasing order, while the magnitude of the eigenvalues represents the function's average variation in each direction. The AS method identifies the active subspace $\tilde{u}$ by partitioning the eigenvalues and eigenvectors as

$$\Lambda = \begin{bmatrix} \Lambda_1 & \\ & \Lambda_2 \end{bmatrix}, \quad W = \begin{bmatrix} W_1 & W_2 \end{bmatrix}, \tag{17}$$



(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 $f = cos(u_1) + exp(-u_2)$ [25]

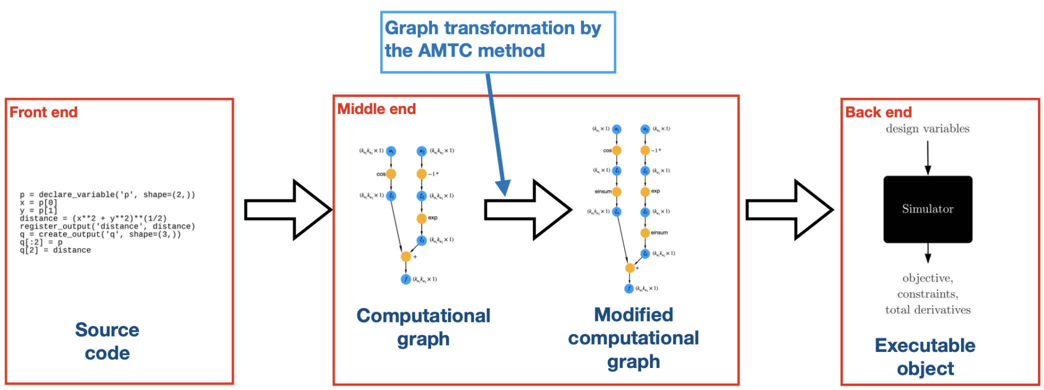


Fig. 3 Demonstration for the implementation of CSDL [25]

where $\Lambda_1 = \text{diag}(\lambda_1, \ldots, \lambda_m)$ and $W_1 \in \mathbb{R}^{m\times d}$ with $m < d$. This results in two sets of rotated coordinates defined as

$$\tilde{u} = W_1^T u, \quad \bar{u} = W_2^T u, \tag{18}$$

such that the function is significantly more variant in the space of $\tilde{u}$ than $\bar{u}$. Once the active subspace is identified, the UQ problem can be solved in this lower-dimensional space, which greatly reduces the computational cost of the analysis.

III. Methodology

In this paper, we first propose a novel framework to connect the integration-based NIPC method to the active subspace approach to solve high-dimensional UQ problems. This involves generating orthogonal PCE basis functions in the active space using the whitening matrix and generating an efficient quadrature rule in the original input space using the designed quadrature method to estimate the PCE coefficients. We then propose an extension of this framework to combine it with the AMTC method to generate the quadrature rule that has a desired tensor structure to take advantage of the sparsity of the computational graph.

A. AS-based NIPC

Consider a high-dimensional UQ problem involving a function,

$$f = \mathcal{F}(u) \quad u \in \mathbb{R}^d, \tag{19}$$

with the indendent uncertain inputs $U = (U_1, \ldots, U_d) \in \mathbb{R}^d$. Assuming d is large enough ($d > 10$), such that directly applying any UQ method would require a large number of model evaluations. We first apply the active subspace (AS) method to find a desired rotation of the input coordinate such that the function is only significantly variant with respect to fewer inputs. This involves first computing the mean squared gradients of the objective function with respect to the uncertain inputs as

$$C = \mathbb{E}[(\nabla_u f)(\nabla_u f)^T], \tag{20}$$

and then an eigenvalue decomposition of it as

$$C = W \Lambda W^T, \quad \Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_d). \tag{21}$$

By inspecting the eigenvalues, the eigenvalues and eigenvectors are partitioned as

$$\Lambda = \begin{bmatrix} \Lambda_1 & \\ & \Lambda_2 \end{bmatrix}, \quad W = \begin{bmatrix} W_1 & W_2 \end{bmatrix} \tag{22}$$

with $\Lambda_1 = \mathrm{diag}(\lambda_1, \ldots, \lambda_m)$, this separates the rotated coordinates into two sets as

$$\tilde{u} = W_1^T u, \quad \bar{u} = W_2^T u, \tag{23}$$

and the function varies significantly more in the active subspace $\tilde{u}$ than the space of $\bar{u}$. Now the function can be written as the rotated inputs as

$$f(\tilde{u}, \bar{u}) = \mathcal{F}(W_1 \tilde{u} + W_2 \bar{u}). \tag{24}$$

Assume we choose m to be reasonably large enough so that the function is nearly invariant in the space of $\bar{u}$, and in the meantime, m is small enough ($m < 10$) so that cost of applying NIPC in the active subspace is affordable. By using the gPC theory and truncating all of the terms of $\bar{u}$, we approximate the model output as

$$F(\tilde{U}) \approx \sum_{i=0}^{q} \alpha_i \Phi_i(\tilde{U}). \tag{25}$$

Since the active subspace,$\tilde{u}$ can be significantly lower dimensional than the original uncertain input space, u, solving PCE coefficients in this equation (25) would require a significantly lower number of model evaluations than applying NIPC with respect to the original uncertain inputs. However, applying the integration-based NIPC method here poses two main challenges.

1) The probability density of the distribution of the uncertain inputs in the active space is unknown which makes it difficult to generate the PCE terms in the active subspace.
2) An efficient quadrature rule in the original input space needs to be derived to calculate the PCE coefficients that correspond to the PCE terms of the active subspace uncertain inputs.

1. Generating PCE basis functions

We solve the first challenge by solving the whitening matrices that represent the coefficients of the PCE basis functions. The rotated coordinates $\tilde{u}$ satisfy a linear relationship with the original coordinates u as $\tilde{u} = W_1^T u$. Since the eigenvectors in W_1 are orthogonal, the active subspace uncertain inputs, $\tilde{U}$, are also mutually independent of each other. This means that the PCE basis functions of $\tilde{u}$ can be formed as a tensor product of univariate orthogonal polynomials in each dimension as

$$\Phi_{\boldsymbol{i}'}(\tilde{u}) = \phi_{i_1}(\tilde{u}_1) \ldots \phi_{i_d}(\tilde{u}_d), \quad |\boldsymbol{i}'| \leq k. \tag{26}$$

The difficulty here is to discover the univariate PCE basis functions for each uncertain input in the active subspace, $\{\phi_i(\tilde{u}_j)\}_{i=0}^{k}$. To compute them, we follow the method in [38] and first represent the univariate PCE basis functions of $\tilde{u}_j$ as

$$\phi(\tilde{u}_j) = \left[\phi_0(\tilde{u}_j), \ldots, \phi_k(\tilde{u}_j)\right]^T = MP_k(\tilde{u}_j), \tag{27}$$

where $P_k(\tilde{u}_j) = [0, \tilde{u}_j, \tilde{u}_j^2, \ldots, \tilde{u}_j^k]^T$ is the monomial basis vector in $\tilde{u}_j$ up to degree k, and $M \in \mathbb{R}^{(k+1)\times(k+1)}$ is a lower triangular matrix storing the coefficients for the minimal basis polynomials in the univariate PCE basis functions. The M matrix is chosen so that the orthogonality property in (5) is satisfied in the $\tilde{u}_j$ dimension as

$$\int_{\Gamma_{\tilde{u}_j}} \phi(\tilde{u}_j)\phi(\tilde{u}_j)^T \rho(\tilde{u}_j) d\tilde{u}_j = I, \tag{28}$$

which can be written as

$$\int_{\Gamma_{\tilde{u}_j}} P_k(\tilde{u}_j)P_k(\tilde{u}_j)^T \rho(\tilde{u}_j) d\tilde{u}_j = M^{-1}M^{-T}. \tag{29}$$

We define the monomial matrix as

$$G := \int_{\Gamma_{\tilde{u}_j}} P_k(\tilde{u}_j)P_k(\tilde{u}_j)^T \rho(\tilde{u}_j) d\tilde{u}_j, \tag{30}$$

then the M matrix satisfies

$$M^{-1}M^{-T} = G, \tag{31}$$

and can be solved through Cholesky factorization of G as

$$M = Q^{-1}, \quad G = QQ^T. \tag{32}$$

However, solving G in (30) would require us to know the probability distribution of the uncertain inputs in the active subspace,$\rho(\tilde{u})$. To address this, we utilize the linear relationship, $\tilde{u}_j = W_j^T u$, where W_j is the $(j+1)$th column of W_1, and apply the change of variables to rewrite (30) as

$$\begin{aligned} G &= \int_{\Gamma_{\tilde{u}_j}} P_k(\tilde{u}_j)P_k(\tilde{u}_j)^T \rho(\tilde{u}_j) d\tilde{u}_j \\ &= \int_{\Gamma_u} P_k(W_j^T u)P_k(W_j^T u)^T \rho(u) du. \end{aligned} \tag{33}$$

The integral in (33) can then be easily computed by Monte Carlo or numerical quadrature approach following the probability density distribution of the original uncertain inputs.

2. Generating an efficient quadrature rule

Using the integration-based NIPC approach, we approximate the PCE coefficients in (25) by solving an integration problem as

$$\begin{aligned} \alpha_i &= \frac{\left\langle F(\tilde{U}), \Phi_i(\tilde{u})\right\rangle}{\left\langle \Phi_i(\tilde{u})^2 \right\rangle} \\ &= \int_{\Gamma_{\tilde{u}}} f(\tilde{u})\Phi_i(\tilde{u})\rho(\tilde{u}) d\tilde{u}. \end{aligned} \tag{34}$$

This results in an integration problem in $\tilde{u}$ space, which is much lower dimensional than the original uncertain input space. However, evaluating this integral is challenging in two parts. Firstly, we cannot directly evaluate $f(\tilde{u})$, the computational model only takes the input of u. For a specific point in $\tilde{u}$ space, there can be an infinite number of points in u that satisfy $\tilde{u} = W_1^T u$. Secondly, the probability density of distribution for the active subspace uncertain inputs, $\rho(\tilde{u})$ is unknown. The objective here is to generate an efficient quadrature rule with the nodes in the original input space.

We denote the quadrature points and the weights as $\boldsymbol{u} := (u^{(1)}, \dots, u^{(n)})$ and $\boldsymbol{w} := (w^{(1)}, \dots, w^{(n)})$ respectively, and the quadrature rule approximates the integral in (34) as

$$\int_{\Gamma_{\tilde{u}}} f(\tilde{u})\Phi_i(\tilde{u})\rho(\tilde{u})d\tilde{u} \approx \sum_{i=1}^{d} w^{(i)} f(u^{(i)})\Phi_i(W_1^T(u^{(i)}). \tag{35}$$

We solve the quadrature rule following the designed quadrature method [21] which was originally derived for solving high-dimensional integration problems. The core idea here is that for an integration problem with respect to the probability density distribution of u, if the nodes and weights of the quadrature rule satisfy the multi-variate moment matching equations as

$$\sum_{i=1}^{n} \Phi_{\boldsymbol{i}'}\left(u^{(i)}\right) w^{(i)} = \begin{cases} 1 & \text{if } |\boldsymbol{i}'| = 0 \\ 0 & \text{for } \alpha \in 0 < |\boldsymbol{i}'| < k, \end{cases} \tag{36}$$

where $\Phi_{\boldsymbol{i}'}$ are the PCE basis functions of u, then the quadrature rule can exactly integration any polynomials of u up to $(2k-1)$th order. In our case, we want to design the quadrature rule with $\boldsymbol{u}$ and $\boldsymbol{w}$ such that after the linear transformation, $\tilde{u} = W_1^T u$, the quadrature rule can exactly integrate polynomials of $\tilde{u}$ to a specific order. Thus the moment-matching equation here is written as

$$\sum_{i=1}^{n} \Phi_{\boldsymbol{i}'}\left(W^T u^{(i)}\right) w^{(i)} = \begin{cases} 1 & \text{if } |\boldsymbol{i}'| = 0 \\ 0 & \text{for } 0 < |\boldsymbol{i}'| < k, \end{cases} \tag{37}$$

where $\Phi_{\boldsymbol{i}'}$ are the PCE basis functions of $\tilde{u}$ that we generated in the previous step. We represent the moment matching equations in (37) as residual equations:

$$\mathcal{R}(\boldsymbol{u}, \boldsymbol{w}) = 0, \tag{38}$$

where $\mathcal{R} : \mathbb{R}^{d\times n} \times \mathbb{R}^n \to \mathbb{R}^n$. The quadrature rule is generated by solving an optimization problem formulated as

$$\begin{aligned} &\min_{\boldsymbol{u},\boldsymbol{w}} \|r(\boldsymbol{u}, \boldsymbol{w})\|_2 \\ &\text{subject to } u^{(j)} \in \Gamma, \quad j = 1, \dots, n, \\ &\qquad\qquad w^{(j)} > 0, \quad j = 1, \dots, n, \end{aligned} \tag{39}$$

which minimizes the norm of the residual equations and constrains the nodes to be within the support range and weights to be positive. The number of nodes to use, n, depends on the dimension of the active subspace, m, and the highest degree of polynomials of PCE basis functions, k, and can be chosen following [26], as

$$n = \begin{cases} \frac{k^m}{m!} & \text{for } m \le 2 \\ 0.9\frac{(2m)^{k-1}}{(k-1)!} & \text{for } m > 2. \end{cases} \tag{40}$$

The optimization problem in (39) can be solved using any common nonlinear optimizer such as the SNOPT (Sparse Nonlinear OPTimizer) [51].

3. Special case: Gaussian uncertain inputs

One special case we can easily generate the PCE basis functions is when all of the uncertain inputs are Gaussian random variables. In this case, the distribution of the uncertain inputs can be written as

$$U \sim \mathcal{N}(\mu, \Sigma), \tag{41}$$

then the active subspace uncertain inputs $\tilde{U}$ are also Gaussian random with distribution

$$\tilde{U} \sim \mathcal{N}(W_1^T\mu, W_1^T\Sigma W_1). \tag{42}$$

In this case, the univariate orthogonal polynomials for every AS uncertain input are Hermite polynomials, and the PCE basis functions are easily formed following (26). For the quadrature rule, we still recommend using the same designed quadrature approach, as this approach results generally require the fewest quadrature points to achieve the same level of accuracy.

4. AS-NIPC algorithm

We refer to this AS-based NIPC method as the AS-NIPC method and present a detailed algorithm in Alg.1. We begin by extracting a lower dimensional active subspace using the AS method as described in equations (20)(21)(22)(23). Next, we generate the univariate PCE basis functions for each AS uncertain input by solving the whitening matrix using equations (33) (32) (27) for a specified polynomial order k. The multivariate PCE basis functions are then formed as a tensor structure of the univariate PCE basis functions using equation (26). Once the PCE basis functions are generated, we choose the number of quadrature points using equation (40), and then form the moment-matching equations in (37). Next, we generate the quadrature rule by solving the optimization problem in (39) using a nonlinear optimizer. Finally, we use the quadrature rule to approximate the PCE coefficients as in equation (35) to perform UQ analysis.

Algorithm 1 AS-NIPC for high-dimensional UQ problems

1: Perform the AS method to compute the active subspace as $\tilde{u} = W_1^T u$
2: Specify the polynomial order,k and generate the PCE basis functions $\Phi_{\boldsymbol{i}'}(\tilde{u})$
3: Choose the required number of quadrature points n
4: Compute the nodes $\boldsymbol{u}$ and weights $\boldsymbol{w}$ in the quadrature rule by solving the optimization problem
5: Evaluate the model on the quadrature points
6: Approximate each PCE coefficient α_i
7: Compute the desired risk measure of the model output

B. AS based NIPC with AMTC

The AS-NIPC method we presented in this paper provides a framework to combine the integration-based NIPC method with the active subspace method to solve high-dimensional UQ problems. However, this method cannot be accelerated by the AMTC method as the quadrature points do not possess a tensor structure. The objective here is to propose an extension of the AS-NIPC method to generate the quadrature points that possess a desired tensor structure such that the model evaluations can be significantly accelerated using the AMTC method. For high-dimensional UQ problems, there often exist some uncertain input operations that affect a small number of operations in the computational graph. In this case, the evaluation cost of the operations that are dependent on these inputs may only take a small percentage (< 10%) of the total model evaluation cost. We refer to these uncertain inputs as sparse uncertain inputs and the other uncertain inputs as non-sparse uncertain inputs. The sparse uncertain inputs can be identified by generating the dependency information of each operation, written as $D(\varphi_i, u_j)$, which stores whether the operation φ_i is dependent on uncertain input u_j as a binary number and the evaluation cost of each operation, written as $O(\varphi_i)$. For example, for the function $f = cos(u_1) + exp(-u_2)$ with the elementary operations defined in (13), we show its dependency information as in Tab. 2. With the graph information, we define the sparsity ratio of an uncertain input as the ratio of its dependent operations' cost over the total model evaluation cost which can be approximated as

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

where $O(f(u))$ is the cost of one model evaluation and the model can be decomposed as a sequence of l operations as $\{\varphi_1, \dots, \varphi_l\}$. We identify the sparse uncertain inputs as its sparsity ratio is < 10% and partition the uncertain inputs as

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

where $u_s = (u_{s_1}, \dots, u_{s_{d_1}}) \in \mathbb{R}^{d_1}$ is the set of sparse uncertain inputs and $u_{ns} \in \mathbb{R}^{d_2}$ 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 can easily form a full-grid quadrature rule following the Gauss quadrature method, such that the quadrature points can be written as

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

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 will only evaluate each operation on the distinct quadrature points in the space of the uncertain inputs on

Table 2 Operations dependency information for function $f = cos(u_1) + exp(-u_2)$ **[25]**

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

which it depends. In the case where all the uncertain inputs are sparse uncertain inputs, the number of evaluations performed on each operation can be significantly lower than the number of quadrature points. However, in most cases, for high-dimensional problems, there often only exist a few sparse uncertain inputs. In this case, we choose the quadrature rule that maintains a tensor structure between the quadrature points in the non-sparse uncertain input space and quadrature points in the space of each sparse uncertain input, such that the quadrature points can be written as

$$\boldsymbol{u} = \boldsymbol{u}^{n_1}_{\{ns\}} \times \boldsymbol{u}^{n_2}_{\{s_1\}} \times \ldots, \boldsymbol{u}^{n_2}_{\{s_{d_1}\}}, \tag{46}$$

where $\boldsymbol{u}^{n_1}_{\{ns\}}$ represents quadrature points in u_{ns} space with n_1 points and $\boldsymbol{u}^{n_2}_{\{s_{d_1}\}}$ represent the quadrature points in u_{s_i} space with n_2 pints. This results in a total of $n_1 n_2^{d_2}$ quadrature points in the quadrature rule however by using the AMTC method to transform the computational graph, the computational cost on the modified computational graph would scale roughly linearly with n_1 as most of the operations in the computational graph only dependent on u_{ns} and will only be evaluated n_1 times. However, for high-dimensional UQ problems, the dimensions for the non-sparse uncertain inputs can still be large enough such that the required model evaluation cost is too much even with the AMTC method. We address this problem by applying the AS-NIPC method within the space of u_{ns} to generate an efficient quadrature rule with respect to the active subspace uncertain inputs. This involves computing the C matrix as

$$C = \mathbb{E}[(\nabla_{u_{ns}} f)(\nabla_{u_{ns}} f)^T] = \int_{\Gamma} (\nabla_{u_{ns}} f)(\nabla_{u_{ns}} f)^T \rho(u) du, \tag{47}$$

then the AS method generates the active subspace inputs as

$$\tilde{u}_{ns} = W_1^T u_{ns}, \tag{48}$$

such that $\tilde{u}_{ns} = W_1^T u_{ns} \in \mathbb{R}^m$ with $m \leq d_1$. Following the AS-NIPC method, we will generate the PCE basis functions $\Phi(\tilde{u}_{ns})$ and the quadrature rule to estimate the PCE coefficients as $\boldsymbol{u}^{\tilde{n}}_{ns}, \boldsymbol{w}^{\tilde{n}}_{ns}$. In this way, the number of quadrature points in u_{ns} is determined by the dimension of the active subspace uncertain inputs rather than the non-sparse uncertain inputs, which means $\tilde{n}$ can be significantly smaller than n_1. Then we can approximate the stochastic function as

$$F(\tilde{U}_{ns}, U_s) \approx \sum_{i=0}^{q} \alpha_i \Phi_i(\tilde{U}_{ns}, U_s), \tag{49}$$

where the PCE basis functions can be constructed as

$$\Phi_i(\tilde{U}_{ns}, U_s) = \Phi_i(\tilde{U}_{ns}) \times \phi(u_{s_1}) \times \ldots \times \phi(u_{s_{d_2}}), \tag{50}$$

where $\phi(u_{s_i})$ are univariate orthogonal polynomials of u_{s_i}. The quadrature rule follows the same tensor structure with the quadrature points as

$$\boldsymbol{u} = \boldsymbol{u}^{\tilde{n}}_{\{ns\}} \times \boldsymbol{u}^{k}_{\{s_1\}} \times \ldots, \boldsymbol{u}^{k}_{\{s_{d_1}\}}, \tag{51}$$

where the $\boldsymbol{u}^{k}_{\{s_i\}}$ is chosen as the k Gauss quadrature points in u_{s_i} dimension. This results in an efficient quadrature rule with a desired tensor structure to use with the AMTC method when it comes to model evaluations.

We refer to this method to combine AS-NIPC with the AMTC method as AS-NIPC-AMTC method and present a detailed algorithm in Alg.2. For a specified computational graph of the model, we first generate the dependency information of each certain input from the computational graph and compute the computational cost of each operation.

Then we identify the sparse uncertain inputs by computing the sparsity ratio of each uncertain input following (43). Next, we separate the uncertain inputs as sparse uncertain inputs u_s and non-sparse uncertain inputs u_{ns}. The next step is to apply the AS-NIPC method on the non-sparse uncertain inputs space to generate the AS PCE basis functions $\Phi_i(\tilde{U}_{ns})$ and the quadrature rule with nodes and weights as $\boldsymbol{u}_{ns}^{\tilde{n}}$ and $\boldsymbol{w}_{ns}^{\tilde{n}}$, respectively. We generate the univariate orthogonal polynomials and Gauss quadrature points for each sparse uncertain input and formulate the PCE basis functions and the quadrature rule following (50) and 51, respectively. Finally, we estimate the PCE coefficients using the quadrature rule and perform the UQ analysis.

Algorithm 2 AS-NIPC-AMTC for high-dimensional UQ problems

1: Specify a computational graph for the model
2: Generate the dependency information from the computational graph $D(\varphi_i, u_j)$
3: Evaluate each operation in the computational graph once and store its computational cost $O(\varphi_i)$
4: Identify the sparse uncertain inputs by computing the sparsity ratio of each uncertain input, $\mathrm{SR}(u_i)$
5: Apply the AS-NIPC method on the non-sparse uncertain inputs space to generate the $\Phi_i(\tilde{U}_{ns})$, $\boldsymbol{u}_{ns}^{\tilde{n}}$ and $\boldsymbol{w}_{ns}^{\tilde{n}}$
6: Formulate the PCE basis functions $\Phi_i(\tilde{U}_{ns}, U_s)$
7: Formulate the quadrature rule in with $\boldsymbol{u}$ and $\boldsymbol{w}$ in a tensor-structure
8: Evaluate the model on the quadrature points using AMTC
9: Compute the PCE coefficients using the quadrature rule
10: Compute the risk measure of QoI

IV. Numerical Results

The test problem here is an 81-dimensional UQ problem involving a lift-plus-cruise electric air taxi adapted from [52]. The representation of the aircraft is shown in Fig. 4. The computational model computes the average ground-level sound pressure level when the aircraft flies along a trajectory, and the UQ problem seeks to find the expectation of the output when the control inputs and acoustic model parameters are subject to uncertainties. The model consists of two sub-models: the trajectory model and the acoustic model. The trajectory model computes the flight trajectory of the aircraft by solving an ODE system given the initial state and control inputs histories which include histories of lift and cruise rotor thrust (x_l x_c). The acoustics model computes the ground-level sound pressure level (SPL) given a flight trajectory with the correlation equation involving three parameters: $(\beta_1, \beta_2, \beta_3)$. The details of the trajectory and acoustic models can be found in [52]. In the UQ problem we have the control inputs with 40 time steps as $x_l = [x_{l_0}, \dots, x_{l_{39}}]$ and $x_c = [x_{c_0}, \dots, x_{c_{39}}]$. We give the history of the average control inputs as $\bar{x}_l = [\bar{x}_{l_0}, \dots, \bar{x}_{l_{39}}]$ and $\bar{x}_c = [\bar{x}_{c_0}, \dots, \bar{x}_{c_{39}}]$ and assume a linear relationship between the control inputs at two consecutive steps and white noise is added at each step, following

$$\begin{aligned} X_{l_{i+1}} &= cX_{l_i} + N_{l_{i+1}}, \quad N_{l_{i+1}} \sim \mathcal{N}(0, 0.003\bar{x}_{l_i}); \\ X_{c_{i+1}} &= cX_{c_i} + N_{c_{i+1}}, \quad N_{c_{i+1}} \sim \mathcal{N}(0, 0.003\bar{x}_{c_i}); \end{aligned} \tag{52}$$

which is a common form to represent the control input uncertainties in motion planning under uncertainty problems [53]. The control inputs histories with their average and 95% confidence interval are shown in Fig. 6. We also set the three parameters in the acoustic model, $(\beta_1, \beta_2, \beta_3)$, as Gaussian random variables. The complete set of 81 uncertain inputs and their distributions are shown in Tab. 3.

This problem is solved by four UQ methods: Monte Carlo, Active subspace-based kriging (AS-kriging), AS-NIPC, and AS-NIPC-AMTC. The Monte Carlo method and the AS-kriging method are chosen as the benchmark methods because Monte Carlo is the optimal method to directly apply to an UQ problem this high dimensional and the AS-kriging method is a popular method to solve high-dimensional UQ via AS approach. These two methods are compared with the two methods we proposed in this paper: AS-NIPC and AS-NIPC-AMTC. Both of the AS-kriging and AS-NIPC methods involve applying the AS method in the original uncertain input space. For both methods, the C matrix in (20) is approximated using the Monte Carlo method with the same 100 sample points. We show the first seven eigenvalues of the C matrix in Tab. 4 and choose $m = 6$ as the dimension of the active subspace. The AS-kriging is implemented following [30] using different numbers of sample points and the AS-NIPC method is implemented following Alg. 1 using different values of k, which correspond to the different maximum polynomial orders for the PCE basis functions.

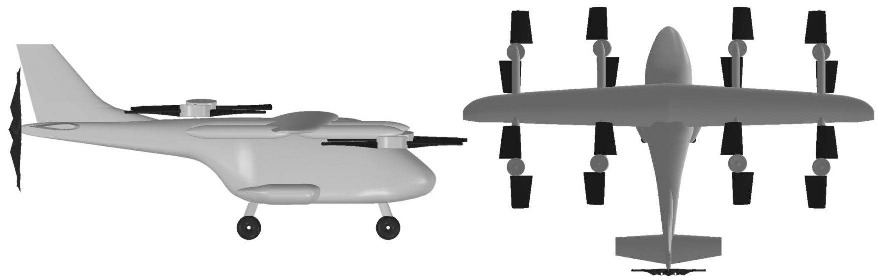
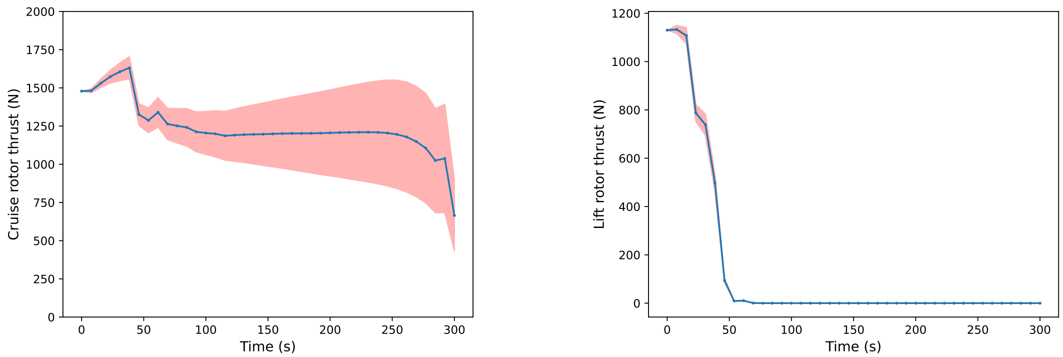


Fig. 4 Representation of the lift-plus-cruise electric air taxi [52]



(a) Cruise rotor thrust **(b) Lift rotor thrust**

Fig. 5 Control inputs with confidence intervals

Table 3 Uncertain inputs in the UQ problem

Uncertain input	Distribution
$N_l \in \mathbb{R}^{39}$	$\mathcal{N}(0, 0.003\bar{x}_l)$
$N_c \in \mathbb{R}^{39}$	$\mathcal{N}(0, 0.003\bar{x}_c)$
$\beta_1 \in \mathbb{R}$	$\mathcal{N}(0.0209, 0.002)$
$\beta_2 \in \mathbb{R}$	$\mathcal{N}(18.2429, 2)$
$\beta_3 \in \mathbb{R}$	$\mathcal{N}(6.729, 0.7)$

For the AS-NIPC-AMTC method, we first compute the sparsity ratio of each uncertain input and the results are shown in Tab. 5. The results show that the parameter uncertain inputs, (β_1,β_2, β_3) are sparse uncertain inputs with a sparsity ratio less than 3%. This is because the parameter uncertainties only affect the operations in the acoustic sub-model whose computational cost takes up only a small amount of the model evaluation. Then the active-subspace method is only applied with respect to the non-sparse uncertain inputs space which includes only N_l and N_c. The first seven eigenvalues are listed in Tab. 4 along with the eigenvalues corresponding to the other two AS methods. From Tab. 4, we observe that the eigenvalues in the AS-NIPC-AMTC decay more rapidly than the other case, and the active subspace dimension is chosen as $m = 5$. We followed Alg. 2 and generated the results of the AS-NIPC-AMTC using the same options of k values as the AS-NIPC method.

The convergence plots of the four UQ methods are shown in Fig. 6, visualizing relative error versus function evaluation cost and another plot versus function evaluation cost plus gradient evaluation cost. The relative error is computed with respect to the Monte Carlo result using 100,000 sample points. For the Monte Carlo and AS-kriging methods, since they involve random sampling, their results are computed as the average relative error over 20 runs. In

Table 4 Normalized eigenvalues (first seven) of the C matrix

AS-kriging & AS-NIPC	AS-NIPC-AMTC
1.0000	1.0000
0.0661	0.0592
0.0649	0.0381
0.0459	0.0080
0.0346	0.0032
0.0194	0.0004
0.0032	0.0002

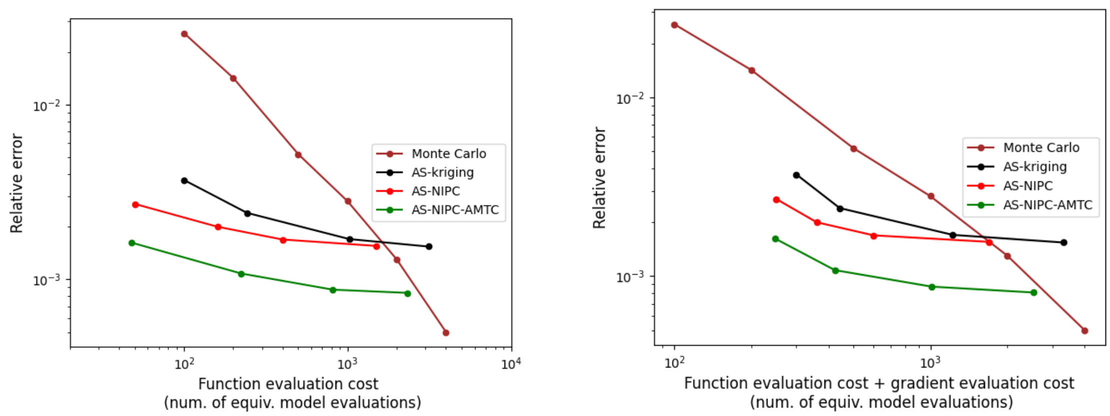
Table 5 Sparsity ratio of the uncertain inputs

Uncertain inputs	Sparsity ratio
N_l	99.31%
N_c	99.32%
β_1	2.02%
β_2	2.03%
β_3	2.14%

Fig. 6a, we report function evaluations cost as only the evaluation cost on the quadrature points. This is particularly useful when we are applying these methods to solve optimization under uncertain problems, as we will only need to apply the AS-NIPC or AS-NIPC-AMTC method once to generate the PCE basis functions and the quadrature rule. Then the same quadrature points are used to repeatedly evaluate the model in every optimization with values of the different design variables. From Fig. 6a we observe that the Monte Carlo method has a higher convergence rate compared with the three AS methods. The Monte Carlo method outperforms the other methods when we can afford 3000 model evaluations. This is because when applying the dimension reduction techniques, as we reduce the dimensions of the problems, we choose to lose some information about the inputs, which will make the dimension reduction-based UQ methods unable to achieve extremely high accuracy. However, if we assume the model is too computationally expensive such that we can only afford much fewer function evaluations (e.g. 100 evaluations), the AS-based methods can achieve a significantly lower relative error compared with the Monte Carlo method. Compared with the AS-kriging method, both of our proposed methods perform better than the AS-kriging method. For 100 function evaluation cost, the relative error of the AS-NIPC is at least 30% lower than AS-kriging, while the AS-NIPC-AMTC is at least 80% lower than AS-kriging. The reason AS-NIPC-AMTC has the best performance is that in the computational graph of the model, there exist three sparse uncertain inputs such that their dependent nodes only takes around 2% of the computational cost. In this case, maintaining a tensor structure of the quadrature rule with each of the sparse uncertain inputs is very effective to use with the AMTC method. Also, by applying the AS method to the uncertain input space excluding the sparse uncertain inputs, we are able to observe a more rapid decaying rate of the eigenvalues for the C matrix as in Tab. 4. This makes us able to choose a smaller dimension for the active sub-space while getting a more accurate reduced-order model. In Fig. 6b, we showed the relative error versus the function evaluation cost plus gradient evaluation cost. This plot is useful to show the efficiency of each method when solving UQ problems. We observe that, with the gradient evaluation cost taken into account, the AS-based methods still perform better than the Monte Carlo when the total cost is less than 1000. We also make similar observations comparing three AS-based methods as they all performed the same number of gradient evaluations when applying the AS method.

V. Conclusion

In this paper, we introduce two UQ methods, AS-NIPC and AS-NIPC-AMTC, for solving high-dimensional UQ problems. The AS-NIPC method connects the integration-based non-intrusive polynomial chaos method to the active subspace (AS) method to efficiently generate the PCE basis functions of the active subspace uncertain inputs and the corresponding quadrature rules to estimate the PCE coefficients. The AS-NIPC-AMTC method extends the AS-NIPC



(a) Function evaluation cost

(b) Function evaluation cost plus gradient evaluation cost

Fig. 6 Convergence plot of the four UQ methods

method by combining it with the graph transformation method, AMTC, to generate a desired tensor structure of the quadrature rule, accelerating the model evaluation cost by using AMTC. We apply both methods to a challenging 81-dimensional UQ problem involving a lift-plus-cruise electric air taxi and compare them with the existing methods. The results show a significant reduction in relative error in the UQ results, particularly when the model evaluation cost is limited to hundreds or fewer evaluations.

One limitation of this work is that AS-NIPC-AMTC method requires the manual identification of sparse uncertain inputs and the formation of quadrature points following the manually set-up tensor structure. However, there may exist better tensor-structure options that are more efficient to use in certain cases. Therefore, future work could extend the AS-NIPC-AMTC method to automatically select the optimal tensor structure option. Additionally, other interesting topics for future research include connecting the graph-accelerated NIPC method to other dimension reduction methods, such as the partial least square method, and investigating the performance of the proposed methods in solving optimization under uncertain problems.

VI. Acknowledgments

The material presented in this paper is, in part, based upon work supported by DARPA under grant No. D23AP00028-00 and by NASA under award No. 80NSSC21M0070

References

[1] Le Maître, O., and Knio, O. M., *Spectral methods for uncertainty quantification: with applications to computational fluid dynamics*, Springer Science & Business Media, 2010.

[2] Montomoli, F., Carnevale, M., D'Ammaro, A., Massini, M., and Salvadori, S., *Uncertainty quantification in computational fluid dynamics and aircraft engines*, Springer, 2015.

[3] 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.

[4] 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.

[5] Albi, G., Pareschi, L., and Zanella, M., "Uncertainty quantification in control problems for flocking models," *Mathematical problems in Engineering*, Vol. 2015, 2015.

[6] 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.

[7] Rockafellar, R. T., Uryasev, S., et al., “Optimization of conditional value-at-risk,” *Journal of risk*, Vol. 2, 2000, pp. 21–42.

[8] Yao, W., Chen, X., Luo, W., Van Tooren, M., and Guo, J., “Review of uncertainty-based multidisciplinary design optimization methods for aerospace vehicles,” *Progress in Aerospace Sciences*, Vol. 47, No. 6, 2011, pp. 450–479.

[9] Beyer, H.-G., and Sendhoff, B., “Robust optimization–a comprehensive survey,” *Computer methods in applied mechanics and engineering*, Vol. 196, No. 33-34, 2007, pp. 3190–3218.

[10] Park, G.-J., Lee, T.-H., Lee, K. H., and Hwang, K.-H., “Robust design: an overview,” *AIAA journal*, Vol. 44, No. 1, 2006, pp. 181–191.

[11] Tu, J., Choi, K. K., and Park, Y. H., “A new study on reliability-based design optimization,” 1999.

[12] Wooldridge, J. M., “Applications of generalized method of moments estimation,” *Journal of Economic perspectives*, Vol. 15, No. 4, 2001, pp. 87–100.

[13] 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.

[14] 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.

[15] 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.

[16] Kaymaz, I., “Application of kriging method to structural reliability problems,” *Structural safety*, Vol. 27, No. 2, 2005, pp. 133–151.

[17] 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.

[18] Rumpfkeil, M. P., “Optimizations under uncertainty using gradients, Hessians, and surrogate models,” *AIAA journal*, Vol. 51, No. 2, 2013, pp. 444–451.

[19] 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.

[20] 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.

[21] 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.

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

[23] Babuška, I., Nobile, F., and Tempone, R., “A stochastic collocation method for elliptic partial differential equations with random input data,” *SIAM Journal on Numerical Analysis*, Vol. 45, No. 3, 2007, pp. 1005–1034.

[24] 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.

[25] Wang, B., Sperry, M., Gandarillas, V. E., and Hwang, J. T., “Accelerating model evaluations in uncertainty propagation on tensor grids using computational graph transformations,” *AIAA Journal*, (submitted).

[26] Wang, B., Orndorff, N. C., and Hwang, J. T., “Optimally tensor-structured quadrature rule for uncertainty quantification,” *AIAA SCITECH 2023 Forum*, 2023, p. 0741.

[27] Morio, J., “Global and local sensitivity analysis methods for a physical system,” *European journal of physics*, Vol. 32, No. 6, 2011, p. 1577.

[28] Abdi, H., and Williams, L. J., “Principal component analysis,” *Wiley interdisciplinary reviews: computational statistics*, Vol. 2, No. 4, 2010, pp. 433–459.

[29] 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.

[30] 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.

[31] Sobol, I. M., "Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates," *Mathematics and computers in simulation*, Vol. 55, No. 1-3, 2001, pp. 271–280.

[32] He, W., Li, G., Zhong, C., and Wang, Y., "A novel data-driven sparse polynomial chaos expansion for high-dimensional problems based on active subspace and sparse Bayesian learning," *Structural and Multidisciplinary Optimization*, Vol. 66, No. 1, 2023, p. 29.

[33] Duong, P. L. T., Yang, Q., Park, H., and Raghavan, N., "Reliability analysis and design of a single diode solar cell model using polynomial chaos and active subspace," *Microelectronics Reliability*, Vol. 100, 2019, p. 113477.

[34] Wiener, N., "The homogeneous chaos," *American Journal of Mathematics*, Vol. 60, No. 4, 1938, pp. 897–936.

[35] 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.

[36] 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.

[37] Rahman, S., "Wiener–Hermite polynomial expansion for multivariate Gaussian probability measures," *Journal of Mathematical Analysis and Applications*, Vol. 454, No. 1, 2017, pp. 303–334.

[38] Lee, D., and Rahman, S., "Practical uncertainty quantification analysis involving statistically dependent random variables," *Applied Mathematical Modelling*, Vol. 84, 2020, pp. 324–356.

[39] 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.

[40] Golub, G. H., and Welsch, J. H., "Calculation of Gauss quadrature rules," *Mathematics of computation*, Vol. 23, No. 106, 1969, pp. 221–230.

[41] 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.

[42] 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.

[43] Blatman, G., and Sudret, B., "Sparse polynomial chaos expansions and adaptive stochastic finite elements using a regression approach," *Comptes Rendus Mécanique*, Vol. 336, No. 6, 2008, pp. 518–523.

[44] Tipireddy, R., and Ghanem, R., "Basis adaptation in homogeneous chaos spaces," *Journal of Computational Physics*, Vol. 259, 2014, pp. 304–317.

[45] Hampton, J., and Doostan, A., "Compressive sampling of polynomial chaos expansions: Convergence analysis and sampling strategies," *Journal of Computational Physics*, Vol. 280, 2015, pp. 363–386.

[46] 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).

[47] Russi, T. M., *Uncertainty quantification with experimental data and complex system models*, University of California, Berkeley, 2010.

[48] Jones, D. R., "A taxonomy of global optimization methods based on response surfaces," *Journal of global optimization*, Vol. 21, 2001, pp. 345–383.

[49] Zhao, M., Alimo, S. R., and Bewley, T. R., "An active subspace method for accelerating convergence in Delaunay-based optimization via dimension reduction," *2018 IEEE Conference on Decision and Control (CDC)*, IEEE, 2018, pp. 2765–2770.

[50] Marzouk, Y. M., Najm, H. N., and Rahn, L. A., "Stochastic spectral methods for efficient Bayesian solution of inverse problems," *Journal of Computational Physics*, Vol. 224, No. 2, 2007, pp. 560–586.

[51] Gill, P. E., Murray, W., and Saunders, M. A., "SNOPT: An SQP algorithm for large-scale constrained optimization," *SIAM review*, Vol. 47, No. 1, 2005, pp. 99–131.

[52] 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.

[53] Janson, L., Schmerling, E., and Pavone, M., "Monte Carlo motion planning for robot trajectory optimization under uncertainty," *Robotics Research: Volume 2*, Springer, 2017, pp. 343–361.