National Technical University of Athens
School of Mechanical Engineering
Fluids Section
Parallel CFD & Optimization Unit

Assessment of Two Sparse Regression PCE Methods for Uncertainty Quantification in Aerodynamics

Master Thesis

Dimitrios Petrou

A thesis submitted in partial fulfillment of the requirements of the Joint Postgraduate Program "Computational Mechanics" of NTUA.

Supervisor: Kyriakos C. Giannakoglou, Professor NTUA

Athens, 2025

Acknowledgements

First and foremost, I owe a great debt of gratitude to my supervisor, Professor Kyriakos Giannakoglou, for the valuable help and support he offered, as well as for the patience he showed me throughout the course of working on this thesis. I can say that the choice of topic was ideal, and I am grateful for the opportunity he gave me to work on such an innovative and interesting subject. I also learned a great deal from his distinctly engineering perspective, which enriched my way of thinking.

I would also like to thank the members of the PCOpt unit of the Fluid Mechanics Section, and especially Dr. Varvara Asouti, for her insightful advice and for the many clarifications and corrections she provided whenever I needed them. In addition, Dr. Evangelos Papoutsis-Kiachagias contributed significantly by helping me address specific problems I encountered.

Furthermore, none of this would have been possible without the support of my parents, Eftychia and Konstantinos, my grandmother Froso, and my sister Chara, who stood by me with understanding and continuously encouraged me throughout this journey.

Finally, I would like to thank Christina for her unwavering support, as well as my friends Christos, Thanos, Christos, and Michalis for their companionship and collaboration. I am also grateful to my classmates, whose cooperation and camaraderie made this period all the more enriching.

Dimitrios Petrou
Athens, September 2025

National Technical University of Athens
School of Mechanical Engineering
Fluids Section
Parallel CFD & Optimization Unit

Assessment of Two Sparse Regression PCE Methods for Uncertainty Quantification in Aerodynamics

Master Thesis
by

Dimitrios Petrou

Supervisor: Kyriakos C. Giannakoglou, Professor NTUA

Athens, 2025

Abstract

This thesis addresses multi-dimensional Uncertainty Quantification (UQ) problems, tested with up to 50 uncertain inputs—by programming and assessing two cost-effective, sparse, regression-based Polynomial Chaos Expansion (PCE) methods: Orthogonal Matching Pursuit (OMP) and the Effective Sampling via Coefficient-Adaptive Polynomial Expansion (ESCAPE). For problems with more than five uncertain inputs, projection methods (even in their sparse variants, like Smolyak grid) and regression-based Non-Sparse PCE (NSPCE) with oversampling ratios (around 3:1) become prohibitive due to their cost. Therefore, this work focuses solely on sparse regression methods, namely OMP and ESCAPE. Both rely on iterative least squares regression, to construct a sparse polynomial basis, by progressively selecting the most relevant polynomials using different selection indicators. This approach allows for undersampling, reducing the number of function evaluations compared to the total number of non-sparse polynomials, while updating their coefficients to minimize the residual error. The two methods are described both theoretically and algorithmically, and the corresponding software has been programmed in C++. OMP sparsifies the polynomial basis by iteratively selecting the most correlated polynomials with the current residual (i.e., the one forming the smallest angle with it), and then orthogonalizing the residual with respect to the selected polynomials. This MSc thesis proposes ESCAPE, a novel sparse PCE method inspired by existing approaches. ESCAPE iteratively builds the sparse polynomial basis by filtering polynomials according to the magnitude of their least squares coefficients, starting from a small initial set of samples and adaptively adding more as needed to ensure a well-posed least-squares problem. To evaluate these methods, they are first tested on two pseudo-engineering problems, followed by three (internal and external) aerodynamic cases, simulated using the OpenFOAM CFD solver.

Εθνικο Μετσοβιο Πολυτεχνειο
Σχολη Μηχανολογων Μηχανικων
Τομεας Ρευστων
Μοναδα Παραλληλης Υπολογιστικης Ρευστοδυναμικης και Βελτιστοποιησης

Αξιολόγηση Δύο Μεθόδων Αραιής Παλινδρόμησης με Χρήση Πολυωνυμικού Χάους για την Ποσοτικοποίηση Αβεβαιότητας στην Αεροδυναμική

Μεταπτυχιακή Εργασία
του

Δημητρίου Πέτρου

Επιβλέπων: Κυριάκος Χ. Γιαννακόγλου, Καθηγητής ΕΜΠ

Αθήνα, 2025

Περίληψη

Η παρούσα διπλωματική εργασία ασχολείται με προβλήματα Ποσοτικοποίησης Αβεβαιότητας (Uncertainty Quantification, UQ) σε πολυδιάστατα συστήματα, μελέτες που περιλαμβάνουν έως και 50 αβέβαιες εισόδους — μέσω υλοποίησης και αξιολόγησης δύο αποδοτικών μεθόδων αραιής παλινδρόμησης βασισμένων σε Ανάπτυγμα Πολυωνυμικού Χάους (Polynomial Chaos Expansion, PCE): την Ορθογώνια Αντιστοίχιση (Orthogonal Matching Pursuit, OMP) και την Αποτελεσματική Δειγματοληψία (αραιού) Πολυωνυμικού Αναπτύγματος Προσαρμοζόμενο μέσω Συντελεστών (Effective Sampling via Coefficient-Adaptive Polynomial Expansion, ESCAPE). Για προβλήματα με περισσότερες από πέντε στοχαστικές εισόδους, οι μέθοδοι προβολής (projection) (ακόμη και σε παραλλαγές αραιού πλέγματος πχ: Smolyak) και η μέθοδος βασισμένου σε παλινδρόμηση, μη-αραιού PCE (Non-Sparse PCE, NSPCE) με λόγους υπερδειγματοληψίας (περίπου 3 : 1) καθίστανται υπολογιστικά απαγορευτικές. Η εργασία αυτή επικεντρώνεται αποκλειστικά σε μεθόδους αραιής παλινδρόμησης, συγκεκριμένα στις OMP και ESCAPE. Και οι δύο βασίζονται σε επαναληπτική παλινδρόμηση ελαχίστων τετραγώνων για την κατασκευή αραιών πολυωνυμικών βάσεων, επιλέγοντας προοδευτικά τα πιο σχετικά πολυώνυμα βάσει διαφορετικών δεικτών επιλογής. Αυτή η προσέγγιση επιτρέπει την υποδειγματοληψία, μειώνοντας τον αριθμό των αξιολογήσεων της συνάρτησης σε σύγκριση με τον συνολικό αριθμό των μη αραιών πολυωνύμων, ενώ ενημερώνει τους συντελεστές τους ώστε να ελαχιστοποιείται το σφάλμα υπολοίπου. Οι μέθοδοι αυτές περιγράφονται τόσο θεωρητικά όσο και αλγοριθμικά, ενώ το αντίστοιχο λογισμικό προγραμματίστηκε σε γλώσσα C++. Η μέθοδος OMP αραιώνει την πολυωνυμική βάση επιλέγοντας επαναληπτικά τα πολυώνυμα που παρουσιάζουν τη μεγαλύτερη συσχέτιση με το τρέχον υπόλοιπο (δηλαδή σχηματίζουν τη μικρότερη γωνία με αυτό), και στη συνέχεια ορθογωνιοποιεί το υπόλοιπο ως προς τα επιλεγμένα πολυώνυμα. Η μεταπτυχιακή εργασία προτείνει τη μέθοδο ESCAPE, μια νέα προσέγγιση εμπνευσμένη από υπάρχουσες τεχνικές. Η ESCAPE κατασκευάζει επαναληπτικά την αραιή πολυωνυμική βάση φιλτράροντας τα πολυώνυμα σύμφωνα με το μέγεθος των συντελεστών τους από τη μέθοδο ελαχίστων τετραγώνων, ξεκινώντας από ένα μικρό αρχικό σύνολο δειγμάτων και προσθέτοντας προσαρμοστικά περισσότερα, εφόσον χρειάζεται, ώστε να εξασφαλιστεί ένα καλά ορισμένο πρόβλημα ελαχίστων τετραγώνων. Για την αξιολόγηση αυτών των μεθόδων, αρχικά δοκιμάζονται σε δύο ψευδο-μηχανολογικά προβλήματα, και στη συνέχεια, σε τρία προβλήματα εσωτερικής και εξωτερικής αεροδυναμικής, τα οποία προσομοιώνονται με το λογισμικό Υπολογιστικής Ρευστοδυναμικής στο περιβάλλον OpenFoam.

Acronyms and Symbols

CFD	Computational Fluid Dynamics
CR	Compression Ratio
CP	Control Point
ESCAPE	Effective Sampling via Coefficient-Adaptive Polynomial Expansion
FFD	Free-Form Deformation
GSA	Global Sensitivity Analysis
LHS	Left-Hand-Side
LOO	Leave-One-Out
NTUA	National Technical University of Athens
NSPCE	Non-Sparse PCE
OLS	Ordinary Least Squares
OMP	Orthogonal Matching Pursuit
PCOpt	Parallel CFD & Optimization Unit
PCE	Polynomial Chaos Expansion
PDF	Probability Density Function
QoI	Quantity of Interest
RHS	Right-Hand-Side
SR	Sampling Ratio
UQ	Uncertainty Quantification

Contents

Chapter 1

Introduction

1.1 Introduction to UQ

Most optimization methods in engineering assume that all input parameters are known with absolute certainty. This approach, known as deterministic optimization, does not account for variability or uncertainty in the input data. Instead, it optimizes the design based on fixed, known conditions, which may include one or more predefined operating points. However, in real problems, both design-related parameters and environmental conditions may vary within a certain range. These variations are referred to as uncertain variables and are typically modeled using statistical distributions—such as normal, Weibull, or others—characterized by their mean values and standard deviations, either known or assumed.

As illustrated in Figure 1.1 [19], the green point denotes the result of deterministic minimization (the optimal design point), whereas the red point indicates the outcome of a minimization that considers uncertainties—commonly referred to as robust optimization (the robust design point). The need to consider the latter is crucial, as the deterministic solution (Figure 1.1) may vary significantly if uncertainty occurs. Real-world problems inherently involve various sources of uncertainty, which must be taken into account during design/optimization.

Therefore, there is a need to develop UQ tools in order to propagate the flow uncertainties from the system input to its output; the output is referred to as the Quantity of Interest (QoI).

UQ requires computing the first two statistical moments of the QoI: the mean and the standard deviation before computing their weighted sum, given by:

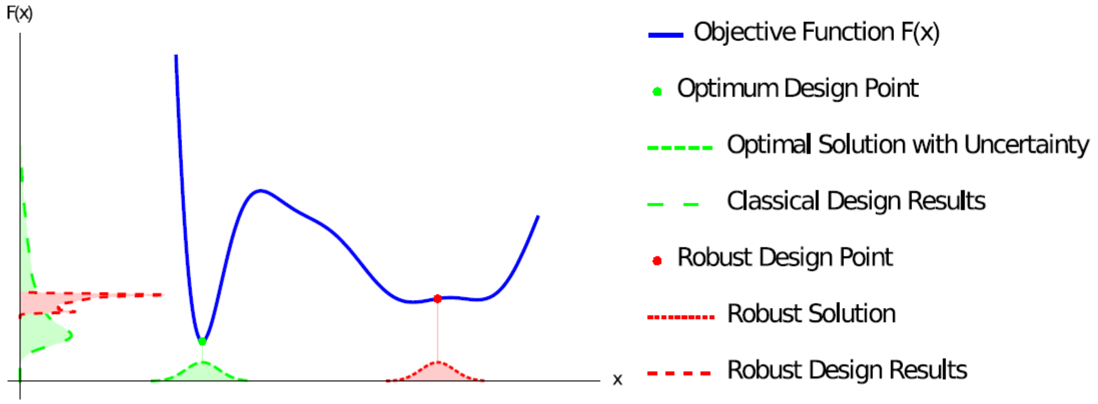

Figure 1.1: *The difference between classical (deterministic) optimization and robust design lies in how uncertainty is handled. Classical optimization minimizes a deterministic objective function, often resulting in solutions sensitive to input uncertainties, causing significant variation in the system response. Robust optimization, by contrast, seeks to minimize a function of the first two statistical moments of the objective function with respect to the uncertain inputs, producing a solution with smaller variability (more stable)[19].*

$$\hat{F} = \hat{\mu}_F + \lambda\hat{\sigma}_F. \tag{1.1}$$

Here, the parameter λ serves as a weighting factor that balances the trade-off between performance (captured by the mean $\hat{\mu}_F$) and robustness (captured by the standard deviation $\hat{\sigma}_F$). $\hat{F}$ becomes the objective function in a Robust Design Optimization (RDO) loop.

In most modern engineering contexts, UQ is becoming an increasingly important field. Deterministic optimization is gradually being replaced by stochastic or RDO to account for inevitable uncertainties in physical phenomena and measurements. However, UQ comes at a high computational cost, especially when dealing with a large number of uncertain variables. A variety of UQ methodologies have been developed, each tailored to meet the specific needs and characteristics of the problem at hand.

It can be implemented using two main categories: intrusive and non-intrusive methods. Intrusive methods typically involve reformulating the governing equations to directly incorporate uncertainties. On the other hand, non-intrusive methods treat the computational model as a black box and can be implemented using either projection-based or regression-based techniques.

This MSc thesis focuses exclusively on non-intrusive, regression-based UQ methodologies, with emphasis on problems characterized by a great number of uncertainties, involving more than 5 and up to 50 uncertain parameters.

1.2 Non-Intrusive UQ Methodology

The two widely used non-intrusive UQ methodologies are projection-based and regression-based methods.

Projection methods compute the polynomial coefficients as a numerical integration problem. Specifically, they rely on the orthogonality of the polynomial basis to project the model response onto each basis function. This projection is typically performed using quadrature nodes, such as Gauss quadrature [12], which approximate the required integrals by function calls at a fixed set of quadrature nodes. The number and location of these nodes are determined by the order of the polynomial expansion and the number of uncertain input variables. While projection methods can be highly accurate for low-dimensional problems, their computational cost grows rapidly with the number of input variables, due to the exponential increase in quadrature nodes required (i.e., the so-called curse of dimensionality) [29, 18, 9]. Even advanced schemes, such as Smolyak sparse grids [1], which exhibit an almost linear growth in function evaluations with dimensionality, become prohibitively expensive for more than five uncertain inputs.

Regression methods aim to approximate a function $F(\mathbf{x})$, where $\mathbf{x}$ represents the uncertain inputs, by expressing it as a linear combination of polynomial basis functions $\psi_j(\mathbf{z})$ [3, 5]. Here, $\mathbf{z}$ denotes a normalized version of $\mathbf{x}$, depending on the distribution type of the inputs. The corresponding polynomial coefficients are determined using the Ordinary Least Squares (OLS) method. This approach computes the coefficients α_j that minimize the deviation between the polynomial approximation and the observed data. A key distinction from projection methods is that, in regression, both the number of evaluation points N and their locations are entirely specified by the user, rather than being determined by the roots of Gauss polynomials (quadrature nodes), as in projection-based approaches. This provides greater flexibility. A general structure of a regression-based UQ algorithm is illustrated in Figure 1.2, showing the workflow from input data and algorithmic parameters to the computation of the mean μ_F and standard deviation σ_F of the QoI.

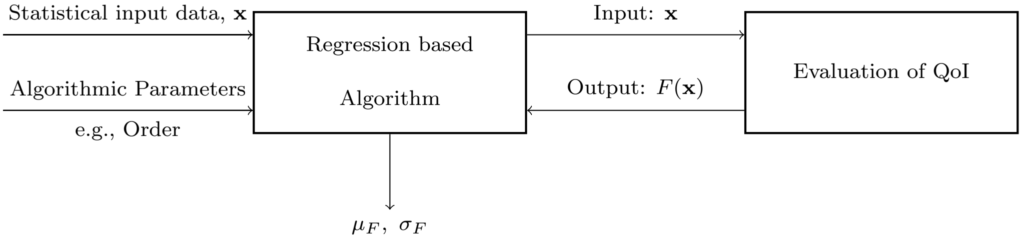

Figure 1.2: *Structure of the regression-based UQ workflow. The UQ algorithm receives algorithm-specific parameters and input data which are evaluated by a black-box model to obtain the corresponding output values $F(\mathbf{x})$. These outputs are used to compute the mean μ_F and standard deviation σ_F of the QoI.*

1.3 Introduction to the PCE

Polynomial Chaos Expansion (PCE) is a statistical tool based on the use of orthogonal polynomials to propagate uncertainty from input stochastic parameters—also referred to as dimensions— to output QoI. It is commonly used to compute statistical moments, such as the mean and standard deviation of the QoI.

PCE was first introduced in 1938 [27]. Initially, the method was limited to stochastic variables that followed a normal distribution, using Hermite polynomials. In 2002, the concept of generalized Polynomial Chaos (gPC) was introduced [29], utilizing the framework proposed in [2] for generating orthogonal polynomials corresponding to different probability distributions. This advancement extended the applicability of PCE to stochastic variables following a wide range of distributions.

A PCE-based method that employs an OLS approach to minimize the deviation between the true model response and its polynomial approximation relies on an adequate number of samples. These samples are typically generated by sampling from distributions appropriate to each input variable. This Monte-Carlo-like sampling approach is well suited for regression-based PCE methods such as OLS, as it provides space-filling coverage of the input space.

To ensure convergence of the regression system, the number of samples N must be sufficiently larger than the total number of polynomial basis terms in a non-sparse (full) basis, denoted as P_{total}. A common way to quantify this requirement is through the Sampling Ratio (SR), defined as

$$SR = \frac{N}{P_{\text{total}}}. \tag{1.2}$$

In practice, for non-sparse systems, values of SR in the range of 2 to 3 are typically required to satisfactorily predict statistical moments. Since $SR > 1$, the above range indicates oversampling. When fewer samples are used, the coefficients are estimated inaccurately. Conversely, using significantly more points than required does not generally improve accuracy but only increases computational cost. Therefore, choosing an appropriate oversampling ratio offers a balanced compromise between efficiency and predictive performance [3].

Nonetheless, even an oversampling ratio of 2–3 highlights the rapid growth of N with dimensionality, as the number of polynomial terms increases combinatorially. This motivates the use of sparse regression techniques in high-dimensional settings, which aim to identify only the most significant polynomial terms while reducing the number of required model evaluations.

1.4 Sparse Regression-Based PCE Methods

Sparse methods are associated with problems involving many uncertain parameters, typically more than five. For instance, in the design of aerodynamic shapes with geometrical imperfections, models often involve a large number of uncertain input parameters. In such cases, sparse regression models are preferred as non-intrusive methods for performing UQ.

As the number of the stochastic inputs (dimensionality) increases, the number of polynomial basis functions in a Non-Sparse PCE (NSPCE) regression system grows rapidly, as illustrated in Figure 1.3. To maintain a sufficient oversampling ratio, the required number of model evaluations increases significantly. This leads to a substantial rise in the cost for computing statistical quantities, such as the mean $\hat{\mu}_F$ and standard deviation $\hat{\sigma}_F$, particularly if each evaluation relies upon computationally expensive models, such as, a solver of the Navier–Stokes equations.

To address these challenges, advanced UQ methodologies have been developed that aim to reduce the number of model evaluations per optimization cycle, while maintaining a high level of accuracy. This thesis focuses on problems involving 8 to 50 uncertain input variables. Due to the prohibitively high computational cost of applying sparse projection methods in such high-dimensional settings, this work exclusively focuses on sparse regression-based approaches. Various sparsification strategies have been proposed in the literature [16] to enable efficient surrogate construction. In this study, two methods are considered: Orthogonal Matching Pursuit (OMP) [4] and Effective Sampling via Coefficient-Adaptive Polynomial Expansion (ESCAPE), a novel method proposed in the context of this MSc thesis and inspired by existing approaches.

In sparse regression-based PCE, the model response is approximated by a truncated polynomial series. The algorithm proceeds in two steps: first, a subset of the most relevant basis functions is selected according to their contribution to the model response; second, the corresponding coefficients are determined via regression. By

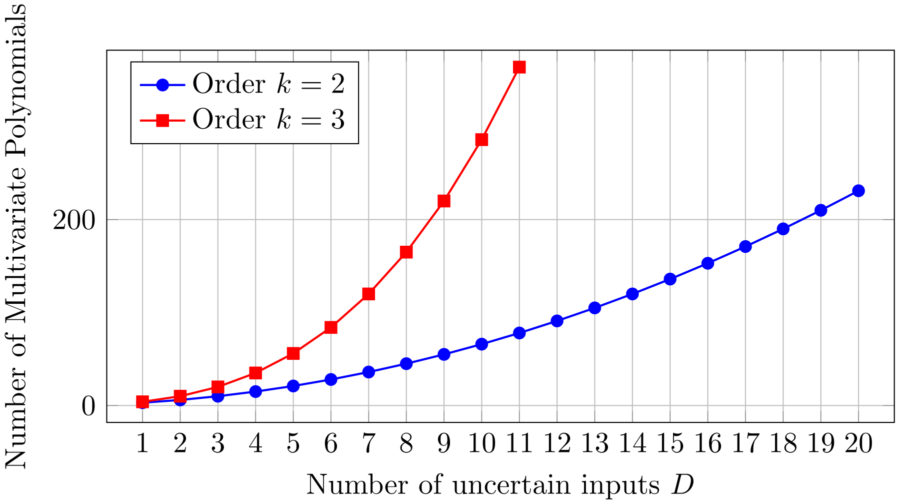

Figure 1.3: *Combinatorial growth in the number of multivariate polynomial basis functions used in NSPCE regression with polynomial orders $k = 2$ and $k = 3$, as the number of uncertain inputs D increases.*

restricting the expansion to only the most relevant basis functions, the number of required model evaluations is significantly reduced, which is essential in high-dimensional problems where each simulation is computationally expensive.

OMP is a greedy algorithm proposed in 1993 [20] as an enhancement to the original Matching Pursuit algorithm [15] introduced the same year. OMP iteratively selects the polynomial basis elements that are most correlated with the current approximation residual, adding them to the sparse (or active) set of regressors. After a basis element has been added to the active set, all corresponding polynomial coefficients are updated via OLS or through recursive formulas. This additional step guarantees that the newly calculated residual is orthogonal to all regressors in the current active set. In each iteration, OMP aims to minimize the discrepancy between the model response and its polynomial approximation [20, 15].

The ESCAPE method relies on the straightforward principle of identifying significant polynomials by examining their coefficients. Additionally, it incorporates the concept of a downward closed basis—discussed in Section 3.1.2—which guarantees a gradual increase in the polynomial order within the basis [14].

Sparse regression PCE methods are widely used today, particularly in problems with many uncertain inputs, due to the significant advantages they offer. One key advantage is their ability to adapt the polynomial basis to the specific characteristics of the problem. By implicitly favoring stochastic input variables that exert greater influence on the QoI, these methods select only a subset of the full polynomial basis having P_{total} elements, namely only P_{sparse} out of them. This results in an expression having fewer terms in the surrogate model. In the general case, P_{sparse} is not known a priori and is computed simultaneously with the corresponding coefficients. A common way to quantify this sparsity of the basis is through the Compression Ratio (CR), defined as:

$$CR = \frac{P_{\text{sparse}}}{P_{\text{total}}} \leq 1. \tag{1.3}$$

This adaptivity allows sparse PCE to maintain comparable accuracy to NSPCE while substantially reducing the number of required model evaluations, especially in problems involving a large number of stochastic inputs.

1.5 Objectives of the Master's Thesis

The primary objective of this Master's Thesis is to explore different approaches for solving sparse UQ problems using sparse regression methods. When the number of input variables exceeds ~ 5, the so-called curse of dimensionality significantly increases the computational cost. To address this challenge, the thesis investigates more economical solution techniques based on sparse formulations of PCE, using sparse regression methods.

Two sparse methods are developed. The first is the well-established OMP. The second is a new method proposed here, called ESCAPE, which draws on ideas from some existing techniques. Each technique builds a sparse polynomial basis using different criteria to reduce the number of required model evaluations.

The two sparse UQ methods are theoretically analyzed and programmed in C++. The methods are first tested on two simplified engineering problems. Then, these are assessed through three applications in internal and external aerodynamics involving the solution of the Navier–Stokes equations for incompressible fluid flows. These applications are carried out using the open-source CFD software OpenFOAM, highlighting the practical applicability and computational performance of the proposed methods.

1.6 Structure of the Master's Thesis

The structure of this thesis is the following:

Chapter 2: Regression PCE, Error Estimation, and Sensitivity Analysis This chapter introduces the theoretical foundations of the PCE method for multidimensional problems. Then, it explores the classical regression approach, exemplified by the OLS problem, and concludes with techniques for error estimation in regression methods and the use of Sobol indices in sensitivity analysis.

Chapter 3: Sparse PCE Methods – Demonstration in Pseudo-Engineering Problems
This chapter focuses on reducing the computational cost of PCE in high-dimensional problems. It presents two sparse regression-based algorithms, namely the OMP and the ESCAPE, which aim to retain accuracy while significantly lowering the number of required model evaluations. Finally, two pseudo-engineering applications are conducted to evaluate and compare the performance of these methods, in problems with 10 and 20 uncertain inputs.

Chapter 4: UQ in Aerodynamic Applications
This chapter applies the OMP and ESCAPE methods, as discussed in the previous chapter, to three aerodynamic cases—both external and internal—featuring 8, 25, and 50 uncertain input parameters, respectively.

Chapter 5: Concluding Remarks and Recommendations for Future Work
This chapter presents the summary of this MSc Thesis and the drawn conclusions, along with a few suggestions for future work.

Appendices
Supplementary material is provided in the appendices. Appendix A presents the fundamental properties of Hermite polynomials, which are essential in the construction of PCE for normally distributed stochastic variables. Appendix B provides a detailed proof of the Leave-One-Out (LOO) error formula used.

Chapter 2

Regression PCE, Error Estimation, and Sensitivity Analysis

This chapter presents the theoretical foundation of the PCE, with a particular focus on regression-based approaches. It begins by formulating the PCE problem as a minimization problem using OLS. Next, it introduces techniques for estimating the approximation error. It concludes with an overview of sensitivity analysis methods within the PCE framework, which are used to quantify the impact of stochastic input variables on the model output.

2.1 PCE in Multidimensional Problems

The theory of PCE [21, 7, 28] suggests that any QoI, F, which depends on a vector of D independent stochastic input variables, can be represented as a series of orthogonal polynomial basis functions multiplied by the PCE coefficients. Let $\mathbf{z} = \{z_0, z_1, \ldots, z_{D-1}\}$ denote a vector of D independent stochastic standardized input variables. This representation enables the approximation of the statistical moments of the output stochastic variable up to chaos order k [19].

Formally, $F(\mathbf{x})$, is approximated as:

$$F(\mathbf{x}) \simeq \hat{F}(\mathbf{x}) := \sum_{j=0}^{P-1} \alpha_j \psi_j(\mathbf{z}), \tag{2.1}$$

where $\psi_j(\mathbf{z})$ denotes the multivariate polynomial basis functions and α_j are the PCE coefficients. In the case of a full tensor product basis up to order k, the total number of basis terms P is given by:

$$P = \frac{(D+k)!}{D!\,k!}. \tag{2.2}$$

Each multivariate basis function $\psi_j(\mathbf{z}_i)$ is the product of univariate orthonormal polynomials $p_{l_d}(z_d)$, where $d \in \{0, \ldots, D-1\}$ denotes the uncertain inputs and $\mathbf{l} = (l_0, l_1, \ldots, l_{D-1}) \in \mathbb{N}_0^D$ is a multi-index that defines the order of each univariate polynomial p_{l_d}:

$$\psi_j(\mathbf{z}) = \psi_j(z_0, z_1, \ldots, z_{D-1}) = \prod_{d=0}^{D-1} p_{l_d}(z_d) \quad j \in \{0, \ldots, P-1\}. \tag{2.3}$$

The total polynomial order for each ψ_j is constrained by:

$$\sum_{d=0}^{D-1} l_d \leq k, \tag{2.4}$$

ensuring that the total order of each multivariate polynomial does not exceed order k. For example, consider the multivariate polynomial basis function with $D = 3$:

$$\psi_4(z_1, z_2, z_3) = p_2(z_1)p_0(z_2)p_0(z_3), \tag{2.5}$$

This is built using a second-order polynomial in z_1, and zeroth-order polynomials (i.e. constants) in z_2 and z_3. Its corresponding multi-index, containing the order of each univariate polynomial, is:

$$\mathbf{I}_4 = \begin{bmatrix} 2 & 0 & 0 \end{bmatrix}. \tag{2.6}$$

The univariate orthogonal polynomials $p_j(z_d)$, with $j \in \{0, \ldots, k\}$ and $d \in \{0, \ldots, D-1\}$, are constructed individually for each input stochastic variable z_d, and satisfy the orthonormality condition:

$$\langle p_m(z_d), p_n(z_d) \rangle = \int_{D_d} p_m(z_d)\, p_n(z_d)\, \omega_d(z_d)\, dz_d = \delta_{mn}, \quad \forall m, n \in \{0, \ldots, k\}, \tag{2.7}$$

where $\omega_d(z_d)$ is the weight function associated with the probability density function (PDF) of z_d, and δ_{mn} is the Kronecker delta:

$$\delta_{mn} = \begin{cases} 0, & m \neq n \\ 1, & m = n \end{cases} \tag{2.8}$$

The classical families of univariate orthonormal polynomials and the corresponding distributions which they are orthonormal to, are summarized in Table 2.1 [26]. Detailed descriptions of these polynomial families, also referred to as the Askey-scheme orthonormal polynomials, can be found in numerous references, such as [29].

Type of variable	**Distribution**	**Orthogonal polynomials**	**Basis**
Uniform	$\frac{1}{2}\mathbf{1}_{[-1,1]}(x)$	Legendre $P_j(x)$	$\frac{P_j(x)}{\sqrt{1/(2j+1)}}$
Gaussian	$\frac{1}{\sqrt{2\pi}}e^{-x^2/2}$	Hermite $He_j(x)$	$\frac{He_j(x)}{\sqrt{j!}}$
Gamma	$x^a e^{-x}\mathbf{1}_{\mathbb{R}_+}(x)$	Laguerre $L_j^a(x)$	$\frac{L_j^a(x)}{\sqrt{\frac{\Gamma(j+a+1)}{j!}}}$
Beta	$\mathbf{1}_{(-1,1)}(x)\frac{(1-x)^a(1+x)^b}{B(a,b)}$	Jacobi $J_j^{a,b}(x)$	$\frac{J_j^{a,b}(x)}{\sqrt{\mathcal{J}_{a,b,j}}}, \quad \mathcal{J}_{a,b,j} = \frac{2^{a+b+1}}{2j+a+b+1}\frac{\Gamma(j+a+1)\Gamma(j+b+1)}{\Gamma(j+a+b+1)\Gamma(j+1)}$

Table 2.1: *Classical families of orthogonal polynomials (subset of the Askey scheme) and their associated orthonormal basis functions.*

Up to this point, the construction of the polynomial basis has been explained. In the following, two approaches for computing the polynomial coefficients in Eq. (2.1) are presented.

2.1.1 Regression Methods

In this work, N samples are selected for the regression process, with $SR > 1$ to ensure that the resulting linear system is overdetermined. To reduce the risk of overfitting, a commonly recommended choice is an oversampling ratio of approximately $SR = 3$.

In regression analysis, the objective is to approximate $F(\mathbf{x})$ by a linear combination of polynomial basis functions $\psi_j(\mathbf{z})$, where the corresponding polynomial coefficients are determined using the OLS method. This approach aims to compute the coefficients α_j that minimize the deviation between the approximation and the model evaluations [12].

Each input sample $\mathbf{x}_i$, representing the i-th realization of the D stochastic input variables with components x_{id}, is derived from a corresponding standardized stochastic vector $\mathbf{z}_i$, whose components z_{id} follow standard distributions (e.g. standard normal), depending on the assumed distribution of each input variable:

$$x_{id} = \mu_d + \sigma_d z_{id}, \quad i \in \{0, \dots, N-1\}, \quad d \in \{0, \dots, D-1\}. \tag{2.9}$$

In the case of normally distributed inputs, the transformation for the first sample $\mathbf{x}_0$ is given by:

$$\mathbf{x}_0 = \begin{bmatrix} x_{0,0} \\ x_{0,1} \\ \vdots \\ x_{0,D-1} \end{bmatrix} = \begin{bmatrix} \mu_0 + \sigma_0 z_{0,0} \\ \mu_1 + \sigma_1 z_{0,1} \\ \vdots \\ \mu_{D-1} + \sigma_{D-1} z_{0,D-1} \end{bmatrix}. \tag{2.10}$$

Here, μ_d and σ_d denote the mean and standard deviation of the d-th stochastic input, respectively, with $d \in D$. For non-Gaussian distributions, however, the mapping from $\mathbf{z}_i$ to $\mathbf{x}_i$ may follow a different transformation rule, depending on the characteristics of the assumed input distribution.

Ordinary Least-Squares

The number of polynomial basis functions P for a given order k and D dimensions is determined by the combinatorial expression of Eq. (2.2).

The problem to be solved using the OLS method, with $\psi_j(\mathbf{z}_i) \in \mathbb{R}^{N \times P}$, $\boldsymbol{\alpha} \in \mathbb{R}^{P \times 1}$, and $F(\mathbf{x}_i) \in \mathbb{R}^{N \times 1}$, where $SR > 1$ (oversampling), is expressed as follows:

$$\begin{bmatrix} \psi_0(\mathbf{z}_0) & \psi_1(\mathbf{z}_0) & \cdots & \psi_{P-1}(\mathbf{z}_0) \\ \psi_0(\mathbf{z}_1) & \psi_1(\mathbf{z}_1) & \cdots & \psi_{P-1}(\mathbf{z}_1) \\ \vdots & \vdots & \ddots & \vdots \\ \psi_0(\mathbf{z}_{N-1}) & \psi_1(\mathbf{z}_{N-1}) & \cdots & \psi_{P-1}(\mathbf{z}_{N-1}) \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_{P-1} \end{bmatrix} = \begin{bmatrix} F(\mathbf{x}_0) \\ F(\mathbf{x}_1) \\ \vdots \\ F(\mathbf{x}_{N-1}) \end{bmatrix}. \tag{2.11}$$

In matrix form, this can be written as:

$$\boldsymbol{\Psi}\boldsymbol{\alpha} = \boldsymbol{f}. \tag{2.12}$$

The left-hand-side (LHS) of Eq. (2.12), $\boldsymbol{\Psi}$, is the polynomial basis evaluated at $\mathbf{z}_i$ (also referred to as the design matrix in the literature), while the right-hand-side (RHS) is the evaluated data $F(\mathbf{x}_i)$. The cost of computing the array on the RHS of Eq. (2.12) is N time-units. The vector $\boldsymbol{\alpha} \in \mathbb{R}^{P \times 1}$ consists of the coefficients that need to be determined.

The OLS system (2.12) is then transformed into:

$$(\mathbf{\Psi}^T\mathbf{\Psi})\boldsymbol{\alpha} = \mathbf{\Psi}^T\boldsymbol{f}, \tag{2.13}$$

computing $\boldsymbol{\alpha}$ as:

$$\boldsymbol{\alpha} = (\mathbf{\Psi}^T\mathbf{\Psi})^{-1}\mathbf{\Psi}^T\boldsymbol{f}, \tag{2.14}$$

provided that $(\mathbf{\Psi}^T\mathbf{\Psi})$ is invertible.

This method computes polynomial coefficients $\boldsymbol{\alpha}$ that minimize the OLS error between the polynomial approximation and the model evaluations.

2.1.2 Calculation of the Statistical Moments

After determining the polynomial coefficients using regression, various statistical quantities can be computed. These include the mean, standard deviation, and higher-order moments such as skewness.

Orthogonal polynomials possess a very useful property that simplifies the calculation of these statistical moments. Specifically, the orthogonality property, which was discussed in the previous subsections, allows for the direct computation of the mean, standard deviation, and higher-order moments.

In this thesis, the first two statistical moments are considered, namely the mean and the standard deviation. The mean of the response is defined as the expectation:

$$\mu_{\hat{F}} = \mathbb{E}[\hat{F}(\mathbf{z})]. \tag{2.15}$$

Substituting the PCE expansion yields:

$$\mu_{\hat{F}} = \sum_{j=0}^{P-1} \alpha_j \, \mathbb{E}[\psi_j(\mathbf{z})] = \alpha_0 \cdot \mathbb{E}[\psi_0] + \sum_{j=1}^{P-1} \alpha_j \, \underbrace{\mathbb{E}[\psi_j(\mathbf{z})]}_{0}. \tag{2.16}$$

By construction, the first basis function is $\psi_0(\mathbf{z}) = 1$, while all higher-order basis functions satisfy:

$$\mathbb{E}[\psi_j(\mathbf{z})] = 0, \quad j \geq 1, \tag{2.17}$$

due to orthogonality. Therefore, only the constant coefficient remains:

$$\mu_{\hat{F}} = \alpha_0. \tag{2.18}$$

The variance of the response is defined as:

$$\sigma_{\hat{F}}^2 = \mathbb{E}\Big[(\hat{F}(\mathbf{x}) - \mu_{\hat{F}})^2\Big]\,. \tag{2.19}$$

Since $\mu_{\hat{F}} = \alpha_0$, the centered expansion becomes:

$$\hat{F}(\mathbf{x}) - \mu_{\hat{F}} = \sum_{j=1}^{P-1} \alpha_j \psi_j(\mathbf{z}). \tag{2.20}$$

Thus, the variance is:

$$\sigma_{\hat{F}}^2 = \mathbb{E}\left[\left(\sum_{j=1}^{P-1} \alpha_j \psi_j(\mathbf{z})\right)^2\right]. \tag{2.21}$$

Expanding the square leads to cross-terms of the form $\alpha_j \alpha_k\, \mathbb{E}[\psi_j(\mathbf{z})\psi_k(\mathbf{z})]$. By orthogonality, all terms with $j \neq k$ vanish, leaving only diagonal contributions:

$$\sigma_{\hat{F}}^2 = \sum_{j=1}^{P-1} \alpha_j^2\, \mathbb{E}[\psi_j^2(\mathbf{z})]. \tag{2.22}$$

If the polynomial basis is orthonormal, i.e. $\mathbb{E}[\psi_j^2(\mathbf{z})] = 1$, the expression simplifies to:

$$\sigma_{\hat{F}} = \sqrt{\sum_{j=1}^{P-1} \alpha_j^2}. \tag{2.23}$$

While validation is important in all UQ methods, it is particularly critical in regression-based approaches due to their data-driven nature and the risk of overfitting or poor generalization from a limited number of samples N. In many practical cases, particularly those involving computationally expensive models, neither analytical solutions nor high-fidelity reference data (e.g. from Monte-Carlo simulations) are available for comparison. To tackle this challenge, various error estimation techniques are introduced and discussed in section 2.2.

2.2 Error Estimation

This section focuses on validation approaches aimed at detecting overfitting and assessing the reliability of the OLS, or surrogate, model. These aspects are crucial not only for ensuring predictive accuracy, but also for reliably estimating the first two statistical moments in the context of UQ. In regression-based PCE, evaluating the generalization capability of the model becomes particularly important when only a limited amount of training data is available.

It is important to note, however, that the validation indices presented in this section do not inherently guarantee the adequacy of the experimental design or the sufficiency of the sample set for reliable uncertainty propagation. Instead, these indices reflect how well the surrogate model reproduces the observed data. A low validation error suggests a good fit within the sampled region, but does not ensure that the surrogate will accurately capture the full variability of the system, particularly in regions of the input space that are underrepresented. To assess the validity of the results presented in this thesis, several evaluation strategies have been explored.

Following the cross-validation technique [25, 11], the initial approach adopted in this work involves selecting a fixed number N of training samples, evaluating them, and using 70% of these to construct the surrogate model while reserving the remaining 30% for validation. This procedure corresponds to a simple holdout validation strategy. If the validation results do not meet the desired accuracy criteria, the number of samples is incrementally increased until satisfactory accuracy is achieved. However, this technique is not practically used in this thesis. As a more economical approach—the Leave-One-Out (LOO) error, ε_{LOO}—was preferred. This method avoids the need to reserve 30% of the samples for validation, thereby making better use of all the evaluated samples.

According to the comparative analysis presented in [17], the LOO error cross-validation technique generally outperforms mean squared error and is the preferred method in this thesis. Let $\hat{F}^{(-i)}$ denote the surrogate model trained on all samples but the i-th sample point, i.e. on the dataset $\mathcal{Z} \setminus \{\mathbf{x}_i\}$, and let F represent the true model. The predicted residual for the i-th observation is defined as the difference between the actual model output at $\mathbf{x}_i$ and the corresponding prediction made by $\hat{F}^{(-i)}$:

$$\Delta^{(i)} = F(\mathbf{x}_i) - \hat{F}^{(-i)}(\mathbf{x}_i). \tag{2.24}$$

The overall LOO error, which serves as an estimate of the model's generalization capability, is given by:

$$E_{\text{LOO}} = \frac{1}{N} \sum_{i=0}^{N-1} \left(\Delta^{(i)}\right)^2 , \tag{2.25}$$

while the normalized LOO error can be expressed as:

$$\varepsilon_{\text{LOO}} = \frac{\sum_{i=0}^{N-1} (\Delta^{(i)})^2}{\sum_{i=0}^{N-1} \left(F(\mathbf{x}_i) - \hat{\mu}_F\right)^2}. \tag{2.26}$$

where $\hat{\mu}_F$ is the sample mean of the model evaluations:

$$\hat{\mu}_F = \frac{1}{N} \sum_{i=0}^{N-1} F(\mathbf{x}_i). \tag{2.27}$$

Conceptually, this method requires training N surrogate models, each leaving out a single data point, and then comparing the prediction at the excluded point with the corresponding true model evaluation. The resulting error estimate provides a reliable measure of the model's predictive performance without the need for additional validation data.

In practice, since the values of $F(\mathbf{x}_i)$ for $i = 0, \ldots, N-1$ are already available from the evaluation of the QoI through CFD simulations, there is no need to explicitly construct N surrogate models to calculate the LOO error. Instead, ε_{LOO} can efficiently be computed from the existing OLS solution without retraining, as described in [5]:

$$\varepsilon_{\text{LOO}} = \frac{\sum_{i=0}^{N-1} \left(\frac{\hat{F}(x_i) - F(x_i)}{1-h_i}\right)^2}{\sum_{i=0}^{N-1} \left(\hat{F}(x_i) - \hat{\mu}_F\right)^2}, \tag{2.28}$$

where h_i is the i-th component of the vector given by:

$$h = \operatorname{diag}\left(\mathbf{\Psi} \left(\mathbf{\Psi}^T \mathbf{\Psi}\right)^{-1} \mathbf{\Psi}^T\right). \tag{2.29}$$

The proof of the equality in Eq. (2.28) can be found in Appendix B.

From this point onward, ε_{LOO}, as defined in Eq. (2.28), will serve as the validation metric for the sparse regression methods employed in this thesis.

2.3 Global Sensitivity Analysis Using Sobol Indices

Global sensitivity analysis (GSA) aims at quantifying the contribution of individual stochastic variables x to a QoI F. GSA can therefore be used, as a prior to UQ step, to help the engineer gain insights about the model at hand and/or screen out unimportant variables before main analysis (UQ). As a result, the dimensionality of the input space is reduced, making the problem more tractable and computationally efficient to solve.

In this thesis, GSA will not be used as an initial assessment step. Instead, in chapter 3, it will serve as a ground truth indicator of how effectively each UQ method captures the importance of the stochastic inputs by examining the polynomial basis constructed by each method.

One of the most widely used techniques for GSA is the Sobol indices, which decompose the output variance into contributions from individual input variables and their interactions. Sobol indices provide both first-order effects (measuring the individual impact of each input) and higher-order effects (capturing interactions between multiple inputs). This variance-based method is model-agnostic, meaning it can be used with any black-box model, and is especially valuable in complex systems where input-output relationships are nonlinear or involve strong interactions [23, 24].

First-Order Sobol Index S_d

The first-order Sobol index quantifies the proportion of the total output variance that can be attributed solely to the variation in the d-th input variable x_d, ignoring any interactions with other inputs. It is defined as:

$$S_d = \frac{\sum_{j \in \mathcal{A}_d} \alpha_j^2}{\sum_{j=1}^{P-1} \alpha_j^2}, \tag{2.30}$$

where $\mathcal{A}_d$ denotes the set of indices for which the corresponding basis function $\psi_j(z_0, \dots, z_{D-1})$ depends only on z_d.

Total Sobol Index S_d^T

The total Sobol index S_d^T measures the overall contribution of the input variable x_d to the output variance, including all possible interaction effects with other inputs. It is defined as:

$$S_d^T = \frac{\sum_{j \in \mathcal{B}_d} \alpha_j^2}{\sum_{j=1}^{P-1} \alpha_j^2}, \tag{2.31}$$

where $\mathcal{B}_d$ denotes the set of indices for which the corresponding basis function $\psi_j(z_0, \dots, z_{D-1})$ depends on z_d, possibly along with other input variables.

Higher-order Sobol indices, which capture interactions between multiple input variables (e.g., second- or third-order), can be derived analogously using the corresponding partial variances. These indices provide a comprehensive view of the input–output dependency structure and are essential for identifying and ranking influential variables in complex systems.

Numerical Example: Sobol Indices for a Simple Model

Consider a QoI $F(\mathbf{x})$ defined as:

$$F(\mathbf{x}) = 10x_1^2 + 0.1x_2 + 0.5x_1x_2, \tag{2.32}$$

where $\mathbf{x} = [x_1, x_2]^T$ and x_1, x_2 are independent and follow a standard normal distribution, i.e., $x_1, x_2 \sim \mathcal{N}(0, 1)$.

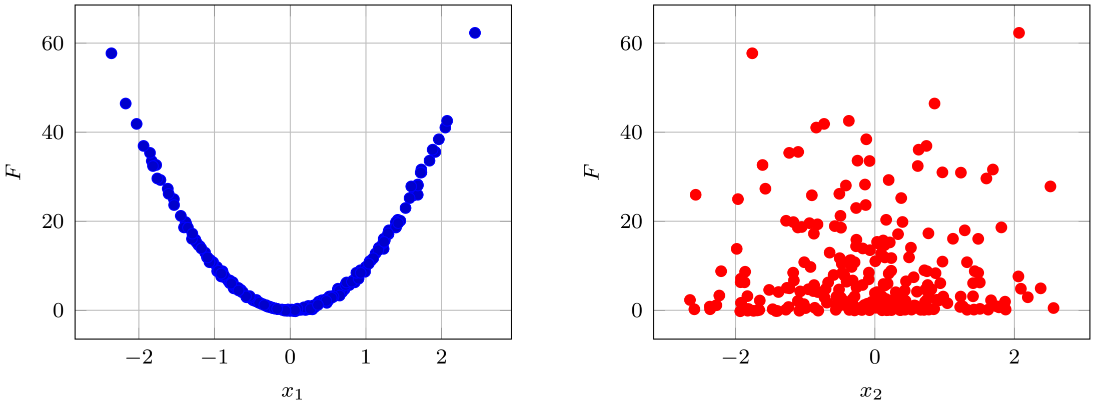

Figure 2.1: *Scatter plots of F, Eq. (2.32) versus x_1 (left) and x_2 (right) for 250 sampled points (x_1, x_2) from their standard normal distributions. In each plot, the other variable is sampled independently, so the scatter reflects the combined influence of both inputs. The left plot shows a clear parabolic trend, indicating that x_1 dominates the variability in F, while the right plot of x_2, shows no obvious trend.*

Figure 2.1 presents scatter plots of QoI in Eq. (2.32) versus the input variables x_1 (left) and x_2 (right) for 250 sampled points (x_1, x_2) drawn independently from standard normal distributions. Each plot represents F as a function of a single input, with the second input sampled independently.

The left plot exhibits a clear parabolic trend, with F increasing as $|x_1|$ grows, indicating that x_1 dominates the variability in F. In contrast, the right plot shows no obvious trend in F, with values scattered around a narrow range. These observations confirm that the variance of F is primarily controlled by x_1 and provide a reference for verifying whether Sobol indices correctly capture the relative importance of each input.

The PCE approximation of $F(\mathbf{x})$ is expressed as:

$$\hat{F}(\mathbf{x}) = \sum_{j=0}^{5} \alpha_j \psi_j(\mathbf{z}),$$

with the following multivariate orthonormal Hermite basis functions:

$$\begin{aligned}
\psi_0(z_1, z_2) &= 1, \\
\psi_1(z_1, z_2) &= z_1, \\
\psi_2(z_1, z_2) &= z_2, \\
\psi_3(z_1, z_2) &= \frac{1}{\sqrt{2}}(z_1^2 - 1), \\
\psi_4(z_1, z_2) &= z_1 z_2, \\
\psi_5(z_1, z_2) &= \frac{1}{\sqrt{2}}(z_2^2 - 1).
\end{aligned}$$

The PCE coefficients $\boldsymbol{\alpha} = [\alpha_0, \alpha_1, \ldots, \alpha_5]^T$, are computed by solving the NSPCE system of Eq. (2.12) via OLS. This system computes coefficients $\boldsymbol{\alpha}$ that best approximate $F(\mathbf{x})$ in the least squares sense.

The resulting PCE coefficients are presented in Table 2.2.

j	**Basis Function** $\psi_j(z_1, z_2)$	**Variables Involved**	**Coefficient** α_j
0	1	—	10
1	z_1	x_1	3.19189×10^{-16}
2	z_2	x_2	0.1
3	$\frac{1}{\sqrt{2}}(z_1^2 - 1)$	x_1	14.1421
4	$z_1 z_2$	x_1, x_2	0.5
5	$\frac{1}{\sqrt{2}}(z_2^2 - 1)$	x_2	-8.88178×10^{-16}

Table 2.2: *PCE basis functions, associated variables, and computed coefficients* α_j *for the QoI defined in Eq.* (2.32).

Based on these coefficients, the statistical moments of the QoI become:

$$\text{Mean} = \alpha_0 = 10.0000000000, \quad \text{Variance} = \sum_{j=1}^{5} \alpha_j^2 = 200.2600000000,$$

$$\text{Standard Deviation} = \sqrt{\sum_{j=1}^{5} \alpha_j^2} = 14.1513250263.$$

The first-order and total Sobol indices are obtained by grouping coefficients according to which variables appear in the basis functions. The terms depending only on z_1 are ψ_1 and ψ_3, those depending only on z_2 are ψ_2 and ψ_5, and the interaction between z_1 and z_2 is captured by ψ_4.

The first-order Sobol indices are computed as:

$$S_1 = \frac{\alpha_1^2 + \alpha_3^2}{\mathrm{Var}[\hat{F}]} = 0.99995, \quad S_2 = \frac{\alpha_2^2 + \alpha_5^2}{\mathrm{Var}[\hat{F}]} = 0.00005,$$

which nearly sum to 1 due to negligible interactions. Similarly, the total Sobol indices are computed as:

$$S_1^T = \frac{\alpha_1^2 + \alpha_3^2 + \alpha_4^2}{\mathrm{Var}[\hat{F}]} = 0.99995, \quad S_2^T = \frac{\alpha_2^2 + \alpha_5^2 + \alpha_4^2}{\mathrm{Var}[\hat{F}]} = 0.00130,$$

which generally do not sum to 1 because they include the contributions of interactions.

These results indicate that the input variable x_1 is by far the dominant contributor to the output variance, while x_2 has only a minor influence. The first-order Sobol indices reflect the variance contribution of each variable alone and nearly sum to 1 due to negligible interactions. The total Sobol indices, which include interaction effects, slightly differ but still confirm that the variance of F is primarily governed by x_1, with x_2 playing a minimal role. These total Sobol indices will be used in Chapter 3.

Chapter 3

Sparse PCE Methods – Demonstration in Pseudo-Engineering Problems

As the number of input stochastic variables D increases, the number of basis terms in the corresponding PCE grows exponentially, making full regression-based approximations both inefficient and costly. This issue is particularly relevant in aerodynamic optimization, where the evaluation model is a CFD tool that numerically solves the Navier–Stokes equations and each simulation can be computationally expensive.

This chapter introduces two sparse regression techniques designed to improve the cost-effectiveness of such approximations without significantly compromising accuracy. Specifically, two different methods for sparsifying the polynomial basis are presented: OMP and ESCAPE.

After presenting the basic concepts and algorithms of both methods, a simple toy example with only two stochastic inputs is solved to illustrate clearly how each method operates. It is important to note that this example is not intended to demonstrate the full capabilities of the algorithms, but rather to introduce their underlying logic. In the last part of this section, two pseudo-engineering problems are solved using the sparse methods.

3.1 Sparse PCE Regression Methods

Sparse regression methods identify the most influential inputs, allowing for a more focused model that prioritizes relevant variables. In this way, the number of samples required for the OLS regression is reduced, which also decreases the number of model evaluations and the associated computational cost.

The uncertain inputs considered in this thesis are assumed to follow a standard normal distribution. The sampling procedure operates as follows: for each of the N samples, a D-dimensional sample is generated, where each component is independently drawn from a standard normal distribution, $\mathcal{N}(0, 1^2)$. To ensure reproducibility, the random number generator is initialized with a fixed seed.

A commonly adopted strategy in sparse PCE regression is to begin with a relatively small number of samples N, aiming to capture the highest possible accuracy by sparsifying the polynomial basis. The QoI is evaluated at these N sample points, and the linear system of Eq. (2.11), or (2.12) is formulated. If the resulting approximation is not satisfactory, the sample size N increases.

The system above can also be written symbolically as:

$$\boldsymbol{\Psi}_{\text{sparse}}\boldsymbol{\alpha}_{\text{sparse}} = \boldsymbol{f}, \tag{3.1}$$

where $\boldsymbol{\Psi}_{\text{sparse}} \in \mathbb{R}^{N \times P_{\text{sparse}}}$, with $P_{\text{sparse}} > N$ and $P_{\text{sparse}} \leq P_{total}$. $\boldsymbol{\Psi}_{\text{sparse}}$ is the sparse LHS, containing evaluations of the polynomial basis at the sample points. The vector $\boldsymbol{\alpha}_{\text{sparse}} \in \mathbb{R}^{P_{sparse}}$ contains the unknown PCE coefficients for the selected bases, and $\boldsymbol{f} \in \mathbb{R}^N$ contains the function evaluations required by the sparse system.

For reasons of space, and since this thesis focuses on sparse regression, whenever reference is made to a sparse PCE method, P_{sparse} will be denoted simply by P, $\boldsymbol{\Psi}_{\text{sparse}}$ by $\boldsymbol{\Psi}$, and $\boldsymbol{\alpha}_{\text{sparse}}$ by $\boldsymbol{\alpha}$. To improve clarity, the main symbols used throughout the text are summarized in Table 3.1.

Symbol	Description
$\mathbf{x}$	Vector of stochastic inputs: $\mathbf{x} = [x_0, \ldots, x_{D-1}]^T$
D	Number of stochastic input variables (dimension) $d_0, \ldots, d_{D-1}$
P	Total number of multivariate orthonormal polynomials (context-dependent)
N	Number of initial samples, indexed by $i \in \{0, \ldots, N-1\}$
P_{total}	Total number of non-sparse polynomials
P_{sparse}	Total number of polynomials after sparsification

Table 3.1: *Summary of notation used in the PCE framework.*

3.1.1 The Orthogonal Matching Pursuit (OMP) Method

The OMP Algorithm is a regression method. It starts with a small, fixed number of evaluated samples N, and $P = 1$, containing only the 0-th order polynomials and iteratively adds only the most significant/correlated polynomials $\boldsymbol{\psi}_j$ to the polynomial basis. This greedy selection continues until at least one out of the three termination criteria is met; these criteria concern the size of the sparse basis that is iteratively being built, the behavior of the LOO error ε_{LOO} through iterations, and the magnitude of the computed correlation index. Upon termination, the coefficients corresponding to the solution at the iteration with the minimum ε_{LOO} are selected as the final result.

In the OMP algorithm, a non-sparse set referred to as the candidate pool $\mathbf{C}$ is used; this contains all the polynomials up to a specified order k, namely: $\boldsymbol{\psi}_j$, $j \in \{0, \ldots, P_{total} - 1\}$. There will also be used a sparse polynomial matrix referred to as the active matrix $\mathbf{A}$, which is gradually filled with significant polynomials during the algorithm. Every polynomial added to $\mathbf{A}$ remains there permanently and is considered part of the sparse set. The number of polynomials in $\mathbf{A}$ is denoted by $P_{\mathbf{A}}$. The notation used is summarized in Table 3.2.

Symbol	Description
$\mathbf{A}$	Active matrix containing $P_{\mathbf{A}}$ selected polynomials
$\mathbf{C}$	Candidate pool containing all P_{total} polynomials up to order k

Table 3.2: *Summary of notation used in the OMP algorithm.*

At each iteration, OMP incrementally chooses and transfers only the most significant polynomial basis function from the candidate pool $\mathbf{C}$ to the active matrix $\mathbf{A}$. The selection is guided by a correlation index, defined as the cosine of the angle between two vectors: the vector of evaluations of each basis polynomial $\boldsymbol{\psi}_j$ over the training inputs, and the residual vector $\mathbf{r}$. The cosine correlation index is computed as follows:

$$\text{Correlation}(\boldsymbol{\psi}_j, \mathbf{r}) = \left|\cos(\boldsymbol{\psi}_j, \mathbf{r})\right| = \left|\frac{\boldsymbol{\psi}_j^T \mathbf{r}}{\|\boldsymbol{\psi}_j\| \, \|\mathbf{r}\|}\right|, \quad j \in \{0, \ldots, P_{total} - 1\} \tag{3.2}$$

Here, $\boldsymbol{\psi}_j \in \mathbb{R}^{N\times 1}$ and $\mathbf{r} \in \mathbb{R}^{N\times 1}$ are column vectors. This index quantifies the alignment between each basis function and the residual, which is defined as:

$$\mathbf{r} = \boldsymbol{f} - \mathbf{A}\boldsymbol{\alpha}, \tag{3.3}$$

guiding the selection of the most relevant polynomial. This selection process continues until a stopping criterion is reached.

Algorithm of OMP

The OMP algorithm begins with an initialization phase that sets the fundamental parameters and defines the problem setup and sampling strategy before the iterative selection of basis functions.

Initialization:

Chaos order selection: Select the chaos order k.

Choosing the number of samples N: The number of samples N can generally be chosen freely. In this thesis, N is set as

$$N = bD \tag{3.4}$$

for this method. The sample-multiplier factor $b = 5$ used in Eq. (3.4) is chosen to reduce overfitting during OLS. A parametric study will be presented in 3.2.1 to justify this choice.

The steps of the OMP algorithm are outlined:

1. Evaluate the model at each of the N samples to obtain $\boldsymbol{f}$:

$$\boldsymbol{f} = \begin{bmatrix} F(\mathbf{x}_0) & F(\mathbf{x}_1) & \cdots & F(\mathbf{x}_{N-1}) \end{bmatrix}^T .$$

2. Compute the number of polynomials P_{total} corresponding to the chaos order k and construct the candidate pool:

$$\mathbf{C} = \begin{bmatrix} \psi_0(\mathbf{z}_0) & \psi_1(\mathbf{z}_0) & \cdots & \psi_{P_{total}-1}(\mathbf{z}_0) \\ \psi_0(\mathbf{z}_1) & \psi_1(\mathbf{z}_1) & \cdots & \psi_{P_{total}-1}(\mathbf{z}_1) \\ \vdots & \vdots & \ddots & \vdots \\ \psi_0(\mathbf{z}_{N-1}) & \psi_1(\mathbf{z}_{N-1}) & \cdots & \psi_{P_{total}-1}(\mathbf{z}_{N-1}) \end{bmatrix}, \tag{3.5}$$

which contains the full set of basis polynomials $\boldsymbol{\psi}_j$, where $j \in \{0, \dots, P_{total}-1\}$ and $i \in \{0, \dots, N-1\}$. This matrix forms the candidate pool, from which polynomials are selected and transferred to the active matrix.

3. Initialize the active matrix $\mathbf{A}$ with only the constant (zeroth-order) polynomial terms $\boldsymbol{\psi}_0$:

$$\mathbf{A} = \begin{bmatrix} \psi_0(\mathbf{z}_0) \\ \psi_0(\mathbf{z}_1) \\ \vdots \\ \psi_0(\mathbf{z}_{N-1}) \end{bmatrix}. \tag{3.6}$$

4. Solve the system:

$$\mathbf{A}\,\boldsymbol{\alpha} = \boldsymbol{f}, \tag{3.7}$$

   with $\boldsymbol{\alpha} = [\alpha_0]$ using the OLS method, and compute the first coefficient α_0. Then, calculate the residual as indicated in Eq. (3.3).

5. Compute the magnitude of cosine correlation index, Eq. (3.2), between the residual vector $\mathbf{r}$ and each candidate polynomial $\boldsymbol{\psi}_j$ vector, i.e. each column vector from the candidate pool $\mathbf{C}$.

6. Select the j-th polynomial vector $\boldsymbol{\psi}_j$ with the highest absolute correlation index, as defined in Eq. (3.2).

7. Copy the selected polynomial from the candidate matrix $\mathbf{C}$, also to the active matrix set $\mathbf{A}$.

8. Update the coefficients α_j of all polynomials in the active set by solving the system in Eq. (3.7), via OLS.

9. Calculate the LOO error (ε_{LOO}) of the system solved.

   - This error serves multiple purposes. First, it can serve as an early stopping criterion to terminate the iterative procedure, as discussed in Step 11b. Second, once the iterative process has concluded, the active polynomial set associated with the lowest ε_{LOO} across all iterations is selected as the optimal sparse basis. Moreover, ε_{LOO} provides a quality indicator for the final approximation. Also, in the case of very simple functions, such as in the example below, ε_{LOO} may even reach zero, thereby directly acting as a termination condition. However, this situation rarely occurs in practice.

10. Update the residual vector $\mathbf{r}$, shown in Eq. (3.3).

11. Repeat steps 5–10 until at least one of the following three criteria is met:

    (a) **Maximum basis size reached:** The size of the active matrix $\mathbf{A}$, with dimensions $N \times P_{\mathbf{A}}$ and $P_{\mathbf{A}} \leq N$ (where $P_{\mathbf{A}}$ increases with each iteration), reaches

$$P_{\mathbf{A}} = \min(P_{\text{total}}, (N-1)/2),$$

which defines the maximum number of possible iterations. The factor $(N-1)/2$ is a practical early stopping criterion: preliminary tests showed that allowing larger active sets substantially increases the computational effort required to solve the OLS systems, without yielding significant improvements in accuracy.

(b) **LOO stagnation:** The LOO error is absolute 0 (e.g. for very simple functions), or it increases continuously, relative to its minimum value reached so far, for a number of iterations equal to at least 10% of the maximum number of iterations, i.e. $\min(P_{\text{total}}, (N-1)/2)$.

(c) **Low correlation with residual:** Terminate if the highest absolute cosine correlation in step 6 falls below 10^{-3}.

If any of these conditions is met, the algorithm terminates. The coefficients corresponding to the iteration with the minimum ε_{LOO}, reached through all iterations, are then selected as the final result. Using these coefficients, the first two statistical moments can be computed.

The OMP algorithm with $N = 5D$ function evaluations is chosen for this thesis, and a parametric study of this choice is presented in Section 3.2.1. To assess its cost reduction compared to non-sparse systems, it can be noted that, while maintaining good accuracy, the computational cost of OMP (shown in Figure 3.1) follows a linear trend. In contrast, the oversampling requirements of NSPCE for $k = 2$, such as the common 3:1 ratio (SR=3), lead to an exponential increase in function evaluations. This significant reduction in function evaluations achieved by OMP makes problems with more than 5 stochastic inputs ($D = 5$) computationally feasible.

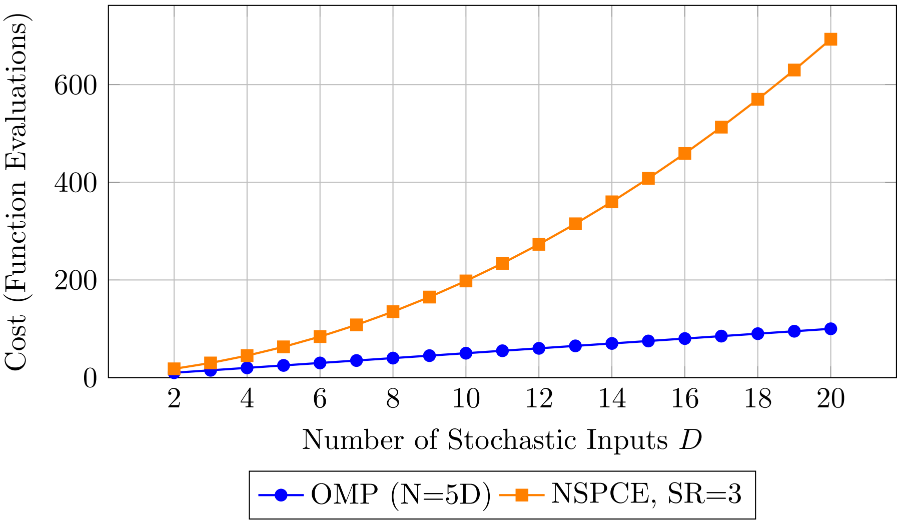

Figure 3.1: *Comparison of the computational cost for constructing polynomial approximations using OMP with $N = 5D$ and NSPCE with $SR = 3$ ($k = 2$) as the number of stochastic inputs D increases. OMP requires substantially less function evaluations.*

Steps 11a, 11b, and 11c serve as early stopping criteria, rather than terminating only when $P_{\mathbf{A}} = \min(P_{\text{total}}, (N-1))$ [16]. These criteria have been observed to significantly reduce the computational time of the OMP iterations. In particular, for systems with up to 50 uncertain inputs, the number of OMP evaluations is reduced substantially without any noticeable loss in accuracy.

Since the LOO error serves not only as a stopping criterion but also as a quality indicator of the regression, it is useful to define a threshold below which the results can be considered reliable upon algorithm termination. Although there is no strict theoretical guarantee, an empirical rule suggests that the LOO error can be considered reliable when:

$$\varepsilon_{\text{LOO}} \leq 10^{-2}. \tag{3.8}$$

In this thesis, ε_{LOO} will be reported alongside the OMP results with Eq. (3.8) in mind, and used as a quantitative metric for evaluating the reliability of the constructed surrogate models. If ε_{LOO} exceeds the acceptable threshold, no additional samples are introduced in this work. Instead, the obtained results should be interpreted with caution, as the predictive accuracy of the sparse polynomial basis may be limited in such cases.

Numerical Example of OMP

Let:

$$F(x_1, x_2) = x_1 + x_2, \tag{3.9}$$

where $x_1 \sim \mathcal{N}(1, 2^2)$ and $x_2 \sim \mathcal{N}(0, 1^2)$ (with $\mathcal{N}(\mu_d, \sigma_d^2)$) are the two ($D = 2$) independent stochastic variables. In closed form formulas, the first and second moments of f are given by:

$$\begin{aligned}
\mu_F &= \mathbb{E}[F] = \mathbb{E}[x_1] + \mathbb{E}[x_2] = 1 + 0 = 1, \\
\sigma_F &= \text{Std}[F] = \sqrt{\text{Var}(x_1) + \text{Var}(x_2)} = \sqrt{2^2 + 1^2} = 2.2360679775.
\end{aligned}$$

In this example, a second-order expansion, $k = 2$, is chosen on purpose to demonstrate how the algorithm handles and rejects second-order polynomial terms for this function. Although the number of samples is generally selected as $N = 5D$ throughout the thesis, a smaller sample size, $N = 5$, is chosen only in this illustrative example to enhance readability and ease of presentation. These samples are evaluated and presented in Table 3.4.

For $k = 2$, the multivariate Hermite polynomials of the i-th sample are listed in Table 3.3.

Each input sample $x_{i,d}$ was generated using a C++ implementation of a standard normal random number generator, as shown in Table 3.4. Each component $z_{i,d}$

Order Sum	Order per dim/on (j_{d_1}, j_{d_2})		Multivariate Hermite Polynomial
0	0	0	$\psi_0(\mathbf{z}_i) = \mathrm{He}_0(z_{i,1})\mathrm{He}_0(z_{i,2}) = \frac{1}{\sqrt{0!0!}}$
1	0	1	$\psi_1(\mathbf{z}_i) = \mathrm{He}_0(z_{i,1})\mathrm{He}_1(z_{i,2}) = \frac{z_{i,2}}{\sqrt{0!1!}}$
	1	0	$\psi_2(\mathbf{z}_i) = \mathrm{He}_1(z_{i,1})\mathrm{He}_0(z_{i,2}) = \frac{z_{i,1}}{\sqrt{1!0!}}$
2	0	2	$\psi_3(\mathbf{z}_i) = \mathrm{He}_0(z_{i,1})\mathrm{He}_2(z_{i,2}) = \frac{(z_{i,2}^2-1)}{\sqrt{0!2!}}$
	1	1	$\psi_4(\mathbf{z}_i) = \mathrm{He}_1(z_{i,1})\mathrm{He}_1(z_{i,2}) = \frac{z_{i,1}z_{i,2}}{\sqrt{1!1!}}$
	2	0	$\psi_5(\mathbf{z}_i) = \mathrm{He}_2(z_{i,1})\mathrm{He}_0(z_{i,2}) = \frac{(z_{i,1}^2-1)}{\sqrt{2!0!}}$

Table 3.3: *Multivariate orthonormal Hermite polynomials $\boldsymbol{\psi}_j$ up to total order $k = 2$ for $D = 2$ stochastic inputs. The second column shows the corresponding multi-index (j_{d_1}, j_{d_2}), while the third one, the normalized product of univariate Hermite polynomials.*

was independently drawn from the standard normal distribution $\mathcal{N}(0, 1^2)$. These samples were subsequently transformed into the physical input space using an affine mapping based on the corresponding mean μ_d and standard deviation σ_d of each input variable, according to Eq. (2.9).

Index i	St/zed Samples $\mathbf{z}_i = (z_{i,1}, z_{i,2})$		Samples $\mathbf{x}_i = (x_{i,1}, x_{i,2})$		Output $F(x_{i,1}, x_{i,2})$
0	1.1679	0.4803	3.3358	0.4803	3.8161
1	-1.1579	0.4344	-1.3158	0.4344	-0.8814
2	-1.1050	-1.3101	-1.2100	-1.3101	-2.5201
3	0.3224	-0.1619	1.6449	-0.1619	1.4830
4	1.0111	0.3828	3.0222	0.3828	3.4050

Table 3.4: *Standardized sample values $\mathbf{z}_i \sim \mathcal{N}(0, 1^2)$ and corresponding stochastic values $\mathbf{x}_i$, along with the output values $F(x_{i,1}, x_{i,2})$ used.*

The non-sparse candidate pool $\mathbf{C}$, constructed by the polynomials $\boldsymbol{\psi}_j$ for $k = 2$, is given by:

$$\mathbf{C} = \begin{bmatrix} \psi_0(\mathbf{z}_0) & \psi_1(\mathbf{z}_0) & \psi_2(\mathbf{z}_0) & \psi_3(\mathbf{z}_0) & \psi_4(\mathbf{z}_0) & \psi_5(\mathbf{z}_0) \\ \psi_0(\mathbf{z}_1) & \psi_1(\mathbf{z}_1) & \psi_2(\mathbf{z}_1) & \psi_3(\mathbf{z}_1) & \psi_4(\mathbf{z}_1) & \psi_5(\mathbf{z}_1) \\ \psi_0(\mathbf{z}_2) & \psi_1(\mathbf{z}_2) & \psi_2(\mathbf{z}_2) & \psi_3(\mathbf{z}_2) & \psi_4(\mathbf{z}_2) & \psi_5(\mathbf{z}_2) \\ \psi_0(\mathbf{z}_3) & \psi_1(\mathbf{z}_3) & \psi_2(\mathbf{z}_3) & \psi_3(\mathbf{z}_3) & \psi_4(\mathbf{z}_3) & \psi_5(\mathbf{z}_3) \\ \psi_0(\mathbf{z}_4) & \psi_1(\mathbf{z}_4) & \psi_2(\mathbf{z}_4) & \psi_3(\mathbf{z}_4) & \psi_4(\mathbf{z}_4) & \psi_5(\mathbf{z}_4) \end{bmatrix}. \qquad (3.10)$$

By substituting the evaluated basis values, $\mathbf{C}$ becomes:

$$\mathbf{C} = \begin{bmatrix} 1.0000 & 0.4803 & 1.1679 & -0.5440 & 0.5610 & 0.2574 \\ 1.0000 & 0.4344 & -1.1579 & -0.5737 & -0.5030 & 0.2409 \\ 1.0000 & -1.3101 & -1.1050 & 0.5066 & 1.4477 & 0.1563 \\ 1.0000 & -0.1619 & 0.3224 & -0.6886 & -0.0522 & -0.6336 \\ 1.0000 & 0.3828 & 1.0111 & -0.6035 & 0.3870 & 0.0158 \end{bmatrix}. \tag{3.11}$$

The active matrix $\mathbf{A}$, which should finally include the selected sparse polynomial basis function, is initialized as just keeping the first column of $\mathbf{C}$, namely:

$$\mathbf{A} = \begin{bmatrix} \psi_0(\mathbf{z}_0) \\ \psi_0(\mathbf{z}_1) \\ \psi_0(\mathbf{z}_2) \\ \psi_0(\mathbf{z}_3) \\ \psi_0(\mathbf{z}_4) \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}. \tag{3.12}$$

In this case, the coefficient vector $\boldsymbol{\alpha}$ contains only a single element α_0, which is computed by solving the system in Eq. (3.7) using OLS. The solution is:

$$\alpha_0 = 1.0604981198.$$

For the computed value of α_0, the residual according to Eq. (3.3) is:

$$\mathbf{r} = \begin{bmatrix} 2.7556 \\ -1.9419 \\ -3.5806 \\ 0.4225 \\ 2.3445 \end{bmatrix}. \tag{3.13}$$

Iteration 1

The correlations are computed according to Eq. (3.2). Specifically, the absolute value of the scalar products between the polynomials $\boldsymbol{\psi}_j$ and the initial residual vector $\mathbf{r}$ are given by:

$$\begin{aligned} |\cos(\boldsymbol{\psi}_0, \mathbf{r})| &= 0.0000 \\ |\cos(\boldsymbol{\psi}_1, \mathbf{r})| &= 0.7227 \\ |\cos(\boldsymbol{\psi}_2, \mathbf{r})| &= 0.9713 \\ |\cos(\boldsymbol{\psi}_3, \mathbf{r})| &= 0.5448 \\ |\cos(\boldsymbol{\psi}_4, \mathbf{r})| &= 0.1937 \\ |\cos(\boldsymbol{\psi}_5, \mathbf{r})| &= 0.1354 \end{aligned}$$

As expected, the residual corresponding to the 0-th order polynomial is, by construction, orthogonal to the new polynomial $\boldsymbol{\psi}_0$. Furthermore, among the candidate basis functions, $|\cos(\boldsymbol{\psi}_2, \mathbf{r})|$ has the highest magnitude, which leads to $\boldsymbol{\psi}_2$ being selected and added to the active matrix $\mathbf{A}$. With the updated $\mathbf{A}$, the goal is to compute the coefficient α_1 while simultaneously updating α_0. The active matrix $\mathbf{A}$ is defined as follows:

$$\mathbf{A} = \begin{bmatrix} \psi_0(\mathbf{z}_0) & \psi_2(\mathbf{z}_0) \\ \psi_0(\mathbf{z}_1) & \psi_2(\mathbf{z}_1) \\ \psi_0(\mathbf{z}_2) & \psi_2(\mathbf{z}_2) \\ \psi_0(\mathbf{z}_3) & \psi_2(\mathbf{z}_3) \\ \psi_0(\mathbf{z}_4) & \psi_2(\mathbf{z}_4) \end{bmatrix} = \begin{bmatrix} 1.0000 & 1.1679 \\ 1.0000 & -1.1579 \\ 1.0000 & -1.1050 \\ 1.0000 & 0.3224 \\ 1.0000 & 1.0111 \end{bmatrix}. \tag{3.14}$$

For clarity, the indices of the coefficients in this thesis refer to their position in the solved system, rather than the order of the corresponding polynomials. Accordingly, the system to solve for $\boldsymbol{\alpha} = [\alpha_0 \quad \alpha_1]^T$, given that $P_{\mathbf{A}} = 2$, is formulated as in Eq. (3.7). After performing OLS, the estimated coefficients are:

$$\begin{aligned} \alpha_0 &= 0.9475861354 \\ \alpha_1 &= 2.3667967354 \end{aligned}$$

ε_{LOO}, is computed according to Eq. (2.28), and is found to be

$$\varepsilon_{\text{LOO}} = 0.1952.$$

In the absence of a termination criterion, the algorithm proceeds to iteration 2.

Though not part of the algorithm itself, the surrogate model constructed till this point is:

$$\begin{aligned} \hat{F}(\mathbf{x}) &= \alpha_0 \psi_0(\mathbf{z}) + \alpha_1 \psi_1(\mathbf{z}) \\ &= 0.9475861354 \frac{1}{\sqrt{0!}} + 2.3667967354 \frac{\frac{x_1 - \mu_1}{\sigma_1}}{\sqrt{1!}} \\ &= -0.2358122323 + 1.1833983677\ x_1. \end{aligned}$$

In Figure 3.2, the original function Eq. (3.9) and the surrogate model $\hat{F}(\mathbf{x}) = -0.2358 + 1.1834\ x_1$ are plotted over the domain $[-3, 3]^2$, illustrating their respective surfaces in the 3D space.

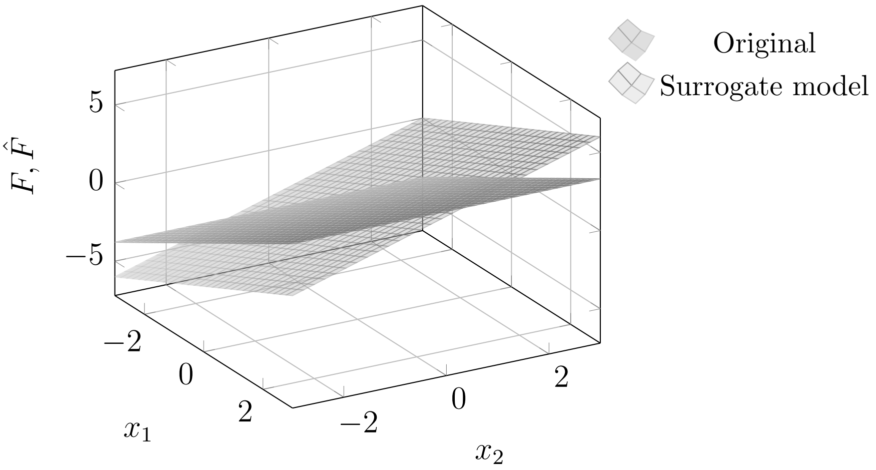

Figure 3.2: *Comparison of the original function in Eq.* (3.9) *(darker surface) and the surrogate model* $\hat{F}(\mathbf{x}) = 1.1834\, x_1 - 0.2358$ *(lighter surface) over the domain* $x_1, x_2 \in [-3, 3]$.

Iteration 2

The residual vector $\mathbf{r}$ is first calculated as:

$$\mathbf{r} = \begin{bmatrix} 0.1044 \\ 0.9115 \\ -0.8524 \\ -0.2278 \\ 0.0643 \end{bmatrix}.$$

The new correlations are:

$$\begin{aligned} |\cos(\boldsymbol{\psi}_0, \mathbf{r})| &= 0.0000 \\ |\cos(\boldsymbol{\psi}_1, \mathbf{r})| &= 0.8388 \\ |\cos(\boldsymbol{\psi}_2, \mathbf{r})| &= 0.0000 \\ |\cos(\boldsymbol{\psi}_3, \mathbf{r})| &= 0.5346 \\ |\cos(\boldsymbol{\psi}_4, \mathbf{r})| &= 0.7468 \\ |\cos(\boldsymbol{\psi}_5, \mathbf{r})| &= 0.2735 \end{aligned}$$

Since the coefficient corresponding to the first-order polynomial is larger, polynomial $\boldsymbol{\psi}_1$ is included in the matrix $\mathbf{A}$. With the updated $\mathbf{A}$, the goal is to compute the coefficient α_2, while simultaneously updating α_0 and α_1. $\mathbf{A}$ is defined as:

$$\mathbf{A} = \begin{bmatrix} \psi_0(\mathbf{z}_0) & \psi_2(\mathbf{z}_0) & \psi_1(\mathbf{z}_0) \\ \psi_0(\mathbf{z}_1) & \psi_2(\mathbf{z}_1) & \psi_1(\mathbf{z}_1) \\ \psi_0(\mathbf{z}_2) & \psi_2(\mathbf{z}_2) & \psi_1(\mathbf{z}_2) \\ \psi_0(\mathbf{z}_3) & \psi_2(\mathbf{z}_3) & \psi_1(\mathbf{z}_3) \\ \psi_0(\mathbf{z}_4) & \psi_2(\mathbf{z}_4) & \psi_1(\mathbf{z}_4) \end{bmatrix} = \begin{bmatrix} 1.0000 & 1.1679 & 0.4803 \\ 1.0000 & -1.1579 & 0.4344 \\ 1.0000 & -1.1050 & -1.3101 \\ 1.0000 & 0.3224 & -0.1619 \\ 1.0000 & 1.0111 & 0.3828 \end{bmatrix}. \tag{3.15}$$

The system for $\boldsymbol{\alpha} = [\alpha_0 \quad \alpha_1 \quad \alpha_2]^T$, with $P_{\mathbf{A}} = 3$, is formulated as in Eq. (3.7) and solved using OLS. The resulting coefficients are:

$$\alpha_0 = 1.0000000000, \quad \alpha_1 = 2.0000000000, \quad \alpha_2 = 1.0000000000.$$

The LOO error is $\varepsilon_{\text{LOO}} = 0.000000$. The second termination criterion is thus activated. For demonstration purposes, however, an additional iteration is carried out to illustrate how the algorithm proceeds.

In this hypothetical third iteration, the residual vector $\mathbf{r}$ becomes zero. As a direct consequence, all cosine correlations with the candidate polynomials are also zero, i.e.,

$$\left|\cos(\boldsymbol{\psi}_j, \mathbf{r})\right| = 0 \quad \forall j.$$

Since the residual vector $\mathbf{r}$ is identically zero, all correlations between the basis vectors $\boldsymbol{\psi}_j$ and the residual also equal zero. This confirms that the current model already took all the 'information' out of $\mathbf{C}$, so the inclusion of additional basis functions offers no further improvement.

The first two statistical moments, computed from the PCE coefficients obtained in iteration 2 (where ε_{LOO} was minimal), are:

$$\begin{aligned} \text{Mean:} \quad & \mu_{\hat{F}} = \alpha_0 = 1.0000000000, \\ \text{Standard Deviation:} \quad & \sigma_{\hat{F}} = \sqrt{\alpha_1^2 + \alpha_2^2} = 2.2360679775, \end{aligned}$$

which match the analytical solution derived above.

The surrogate model created using the same coefficients is:

$$\begin{aligned} \hat{F}(\mathbf{x}) &= \alpha_0\psi_0(\mathbf{z}) + \alpha_1\psi_2(\mathbf{z}) + \alpha_2\psi_1(\mathbf{z}) \\ &= \frac{1}{\sqrt{0!}} + 2\frac{\frac{x_1-\mu_1}{\sigma_1}}{\sqrt{1!}} + \frac{\frac{x_2-\mu_2}{\sigma_2}}{\sqrt{1!}} \\ &= x_1 + x_2 = F(\mathbf{x}). \end{aligned}$$

which matches the QoI.

The reliability of OMP results with $N = 5D$ samples is assessed after the algorithm has terminated, ideally with $\varepsilon_{\text{LOO}} < 0.01$. If this threshold is exceeded, a common approach—though not illustrated in this thesis—is to add additional samples ($N_{\text{new}} > N$) and rebuild the polynomial basis from scratch. To implement a similar strategy automatically, without user intervention, and using a different sparsification criterion, the ESCAPE algorithm is introduced.

3.1.2 The Effective Sampling via Coefficient-Adaptive Polynomial Expansion (ESCAPE) Method

To obtain results through a robust determination of the sample size N, the ESCAPE method is proposed. ESCAPE uses a different sparsification index than OMP and is capable of dynamically adding samples during the algorithm's iterations, while ensuring that the number of samples is sufficient relative to the current sparse polynomial basis at each iteration, as dictated by the $m : 1$ sample-to-polynomial ratio.

This method always starts with the smallest possible chaos order $k_{\text{curr}} = 1$, that will be increased by one in every iteration until the final order k_{final} is reached (e.g. for $k_{\text{final}} = 2$ ESCAPE will conduct 2 iterations). Initially, a small number of samples $N = 3D$ is evaluated. At each iteration, the method expands the sparse polynomial basis by adding only the most significant polynomials from a downward closed set of polynomials of order $k \leq k_{\text{curr}}$. A sparsification index is used to identify these polynomials, based on the magnitude of the associated coefficients—also referred to as sensitivities, as they reflect the influence of each basis function on the model output. ESCAPE may adaptively enrich the sample set during each iteration, ensuring that the number of samples remains sufficient relative to the current sparse polynomial basis, providing stable and accurate OLS regression.

Analogous to OMP, ESCAPE operates with an active matrix $\mathbf{A}$ that finally collects the sparse polynomial basis. Every polynomial added to $\mathbf{A}$ remains there permanently and is considered part of the sparse set. A matrix called $\mathbf{S}$ is introduced solely to solve the least squares problem; it does not carry any physical significance and is used primarily for explanatory purposes.

This method provides a conceptual framework for identifying significant polynomial terms via their coefficients within regression-based UQ, drawing inspiration from recent developments such as [14]. The concept of a downward closed polynomial basis—adapted from [14] and incorporated within the ESCAPE framework—is employed to structure the polynomial basis effectively.

In brief, this concept allows the sparse polynomial basis for the d-th dimension, with $d \in \{0, \dots, D-1\}$, to include a univariate polynomial $p_{l_d}(z_{id})$ of order $j_{\text{new}} \in \{0, \dots, k\}$ with $i \in \{0, \dots, N-1\}$ only if all lower-order polynomials $p_{l_d}(z_{id})$ with $l_d < j_{\text{new}}$ are already present. This ensures a hierarchical, nested structure, preventing the inclusion of higher-order terms without their corresponding lower-order terms. Mathematically, it has been demonstrated that employing a downward closed polynomial basis improves accuracy [6].

This concept is implemented within the software developed for the ESCAPE method, as detailed below.

Downward Closed Polynomial Spaces

In multidimensional PCE, a multivariate polynomial basis is constructed from tensor products of univariate orthogonal polynomials. The corresponding multi-index set defines the structure of the expansion. These univariate indices, as introduced in Chapter 2, are now assembled into a multidimensional structure called Λ, as shown below.

For example, the following j-th basis function with 3 inputs:

$$\psi_j(z_{i,1}, z_{i,2}, z_{i,3}) = p_1(z_{i,1})\, p_0(z_{i,2})\, p_2(z_{i,3}), \quad \text{with } i \in \{0, \ldots, N-1\},$$

correspond to the multivariate index $\Lambda = \{(1,0,2)\}$ which is the product of a first-degree polynomial in $z_{i,1}$, a zero-degree polynomial in $z_{i,2}$, and a second-degree polynomial in $z_{i,3}$.

At each iteration of ESCAPE, a candidate set of new polynomial terms is generated and denoted as Λ^+. These candidates are then filtered to ensure the downward closed property is preserved. The resulting admissible subset is denoted by $\Lambda^+_{\text{adm}} \subseteq \Lambda^+$, containing only those multi-indices whose corresponding lower-order terms are already included in the current sparse basis. For example:

In a problem with two stochastic inputs, consider that the multi-index set of the sparse basis at a given iteration is:

$$\Lambda = \{(0,0),\ (0,1)\}.$$

which is illustrated by the green nodes in Figure 3.3. Due to the increment of the current chaos order by one, from $k_{\text{curr}} = 1$ to $k_{\text{curr}} = 2$, the algorithm proposes a set of candidate multi-indices to be added (indicated in yellow in Figure 3.3):

$$\Lambda^+ = \{(1,0),\ (0,2),\ (1,1),\ (2,0)\}.$$

Multi-indices $(1,1)$ and $(2,0)$ are not admissible since not all their lower-order predecessors are present in Λ. Specifically, $(1,1)$ requires $(1,0)$, and $(2,0)$ requires

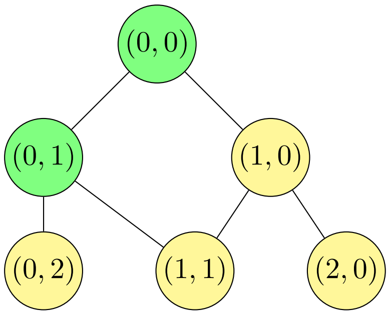

Figure 3.3: *Illustration of downward closed multi-index selection. The root node* $(0,0)$ *and its immediate successor* $(0,1)$ *are already in the sparse basis (green). Candidate multi-indices (yellow) include* $(1,0)$, $(0,2)$, $(1,1)$, *and* $(2,0)$. *Nodes* $(1,1)$ *and* $(2,0)$ *are rejected due to missing predecessors in the index set, while* $(1,0)$ *and* $(0,2)$ *are accepted and added to the admissible set, updating the selection.*

$(1,0)$, which is missing from Λ. Therefore, both $(1,1)$ and $(2,0)$ are rejected. The admissible subset is thus:

$$\Lambda^+_{\text{adm}} = \{(1,0),\ (0,2)\}.$$

This leads to the extended index set:

$$\Lambda_{\text{ext}} = \Lambda \cup \Lambda^+_{\text{adm}} = \{(0,0),\ (0,1),\ (1,0),\ (0,2)\}.$$

From this extended set, the most suitable candidate—according to a predefined selection criterion—is incorporated into the basis. This process effectively acts as another 'filter' (apart from the sensitivity based one), preventing non-admissible terms from entering the polynomial basis.

The notation used in ESCAPE is summarized in Table 3.5.

Symbol	Description
$\mathbf{A}$	Active matrix containing $P_{\mathbf{A}}$ selected polynomials
$\mathbf{S}$	Matrix used exclusively for solving OLS; initially empty
Λ	Multi-index set defining the multivariate polynomial basis
Λ^+	Candidate multi-indices proposed for addition to the basis
Λ^+_{adm}	Admissible subset of Λ^+ satisfying downward-closed property

Table 3.5: *Summary of notation used in the ESCAPE algorithm and multidimensional PCE.*

Algorithm of ESCAPE

The initialization phase of the ESCAPE method primarily sets the final chaos order.

Initialization:

Chaos order selection: Select the final order k_{final} which, in the context of this algorithm, also corresponds to the number of iterations.

The steps of the ESCAPE are outlined:

1. Set the current order $k_{\text{curr}} = 1$.

2. Predefine an initial number of samples, for instance $N = 3D$, which may later be enriched with additional samples. Evaluate the model at these samples to obtain

   $$\boldsymbol{f} = \begin{bmatrix} F(\mathbf{x}_0) & F(\mathbf{x}_1) & \cdots & F(\mathbf{x}_{N-1}) \end{bmatrix}^T ,$$

   and calculate the initial active matrix $\mathbf{A}$, which at first contains only the constant polynomial $\boldsymbol{\psi}_0$:

   $$\mathbf{A} = \begin{bmatrix} \psi_0(\mathbf{z}_0) \\ \psi_0(\mathbf{z}_1) \\ \vdots \\ \psi_0(\mathbf{z}_{N-1}) \end{bmatrix} . \tag{3.16}$$

   This matrix is subsequently updated at each iteration by adding the most significant polynomial identified by the selection criterion.

3. Copy all polynomial terms from the $\mathbf{A}$ matrix into the $\mathbf{S}$ matrix.

4. Add to the $\mathbf{S}$ matrix all the polynomials of order up to the current k_{curr} , unless they are already present in $\mathbf{S}$ (from the previous step). Mark the polynomials added only in this step— with indices like $\boldsymbol{\psi}_{j_{\text{new}}}$—to distinguish them from the existing ones.

   Note: previously rejected polynomials due to admissibility or sensitivity are not automatically excluded at this stage; they may be reconsidered if they now satisfy the criteria.

5. Check the marked polynomials $\boldsymbol{\psi}_{j_{\text{new}}}$ in the $\mathbf{S}$ matrix for admissibility. Remove the non-admissible ones.

   - This approach is based on the assumption that the set of multi-indices associated with the polynomial space is downward closed, as described in Section 3.1.2. If this condition is not met, the corresponding multi-indices are rejected.

   - The admissible polynomials that remain in $\mathbf{S}$ are marked as $\boldsymbol{\psi}_{j_{\text{new/adm}}}$.

6. Sample addition and function evaluation step (if needed): Check whether the current number of samples N is sufficient to support the number of candidate polynomials in matrix $\mathbf{S}$, denoted $P_{\mathbf{S}}$, according to Eq. (3.17).

- Verify whether the following inequality is satisfied:

$$N \geq 2P_{\mathbf{S}}, \tag{3.17}$$

  which ensures that the least-squares system, Eq. (3.19), has a sufficient number of samples compared to the number of polynomials.

- If the inequality is not satisfied, increase N by adding enough samples so that

$$N = mP_{\mathbf{S}}. \tag{3.18}$$

  Evaluate the model at the added samples. The updated vector of model evaluations is then

$$\boldsymbol{f} = \begin{bmatrix} F(\mathbf{x}_0) & F(\mathbf{x}_1) & \cdots & F(\mathbf{x}_{N-1}) \end{bmatrix}^T,$$

  which now includes all current samples.

In Eq. (3.18), $m = 2$ will be used, and a parametric study will be conducted later to justify why the sample-multiplier factor $m = 2$ is optimal.

7. Solve the System:

$$\mathbf{S}\boldsymbol{\alpha} = \boldsymbol{f}, \tag{3.19}$$

using OLS and compute the coefficients $\boldsymbol{\alpha} = [\alpha_0, \dots, \alpha_{P_{\mathbf{S}}-1}]^T$.

8. Compute the sensitivity index, as defined in Eq.(3.20), for the coefficients $a_{j_{\text{new/adm}}}$ that are related with the $\boldsymbol{\psi}_{j_{\text{new/adm}}}$ in $\mathbf{S}$. Permanently transfer only the significant polynomials of them (according to Eq.(3.21)) in $\mathbf{A}$ matrix and reject the rest.

- Compute the sensitivity index for each newly calculated coefficient $a_{j_{\text{new/adm}}}$, associated with $\boldsymbol{\psi}_{j_{\text{new/adm}}}$ in $\mathbf{S}$:

$$\eta_{j_{\text{new/adm}}} = \alpha^2_{j_{\text{new/adm}}}. \tag{3.20}$$

- Compute the mean sensitivity index η_{mean} from the previously computed values $\eta_{j_{\text{new/adm}}}$

- Compare each computed sensitivity index with the threshold:

$$\eta_{j_{\text{new/adm}}} > \frac{\eta_{\text{mean}}}{2}. \tag{3.21}$$

If this condition holds, transfer the corresponding polynomials to the active matrix $\mathbf{A}$.

- Upon completion of this step, empty the $\mathbf{S}$ matrix.

9. Increase the current order by one, i.e. set $k_{\text{curr}} = k_{\text{curr}} + 1$.

10. Repeat steps 3–9 until the iteration with $k_{\text{curr}} = k_{\text{final}}$ has been completed.

    After termination, the sparse basis $\mathbf{A}$ has been constructed. Solve

$$\mathbf{A}\boldsymbol{\alpha} = \boldsymbol{f}, \tag{3.22}$$

using OLS to obtain the final coefficients α_j. Finally, compute ε_{LOO}, which serves as a quality indicator of the expansion. Using these coefficients, the first two statistical moments can be computed.

A graph similar to Figure 3.1 illustrating the computational efficiency of the ESCAPE method would, unfortunately, be inaccurate, since the number of function evaluations selected by ESCAPE is a priori unknown and case-dependent. Nevertheless, this characteristic underlines the algorithm's ability to possibly reduce the linear cost observed in certain OMP scenarios by employing an adaptive strategy, where function evaluations are performed according to the sampling requirements of the sparse OLS system (as described in Step 6). In some cases, the OMP algorithm may overestimate the required samples (e.g., when $N = 5D$ proves excessive), ESCAPE may achieve a lower computational cost; in other cases, the cost may be comparable or slightly higher.

As seen in the algorithm, k_{final} determines both the number of iterations performed and the maximum possible chaos order in the basis. For example, setting $k_{\text{final}} = 3$ does not guarantee the inclusion of polynomials of degree three, since such terms may repeatedly be rejected due to the downward-closed condition or the sparsification criterion. Instead, it merely allows for their potential inclusion. So, increasing k_{final} raises the maximum admissible chaos order, which also increases the computational cost as the polynomial basis grows.

It is also important to note that the admissibility basis check is performed prior to the sample addition step. This ensures that the number of basis polynomials is minimized as much as possible before potentially adding new samples, as dictated by Eq. (3.18).

Numerical Example of ESCAPE

Let the same QoI as in Eq. (3.9) be considered, where $x_1 \sim \mathcal{N}(1, 2^2)$ and $x_2 \sim \mathcal{N}(0, 1^2)$ are uncertain inputs.

Initially, the order of the PCE is selected, and here $k_{\text{final}} = 2$ is purposely chosen, as before, to demonstrate how the algorithm handles and rejects second-order polynomial terms for this function. The number of samples is adjusted by the algorithm. Initially, $N = 3D = 6$ samples are selected. The multivariate polynomials $\boldsymbol{\psi}_0$ are computed as shown in Table 3.3.

The input samples $\mathbf{z}_i$, where $i \in \{0, \ldots, N-1\}$, are generated as in the previous example using a standard normal random number generator, see Table 3.6.

Index i	**St/zed Samples** $\mathbf{z}_i = (z_{i,1}, z_{i,2})$		**Samples** $\mathbf{x}_i = (x_{i,1}, x_{i,2})$		**Output** $F(x_{i,1}, x_{i,2})$
0	1.1679	0.4803	3.3358	0.4803	3.81615
1	-1.1579	0.4344	-1.3158	0.4344	-0.881432
2	-1.1050	-1.3101	-1.2100	-1.3101	-2.52014
3	0.3224	-0.1619	1.6449	-0.1619	1.48296
4	1.0111	0.3828	3.0222	0.3828	3.40496
5	0.7779	0.7266	2.5557	0.7266	3.28235

Table 3.6: *Set of $N = 6, D = 2$ standardized values $\mathbf{z}_i \sim \mathcal{N}(0, 1^2)$, corresponding stochastic values $\mathbf{x}_i$, and outputs $F(x_{i,1}, x_{i,2})$ used for regression.*

The active matrix $\mathbf{A}$, filled initially with only the 0-th order polynomial basis functions computed at the sample points, is initialized as:

$$\mathbf{A} = \begin{bmatrix} \psi_0(\mathbf{z}_0) = 1 \\ \psi_0(\mathbf{z}_1) = 1 \\ \psi_0(\mathbf{z}_2) = 1 \\ \psi_0(\mathbf{z}_3) = 1 \\ \psi_0(\mathbf{z}_4) = 1 \\ \psi_0(\mathbf{z}_5) = 1 \end{bmatrix}. \tag{3.23}$$

The polynomial index parameter Λ lists the polynomial orders that exist for each dimension (column) in order to check for admissibility later on. For the $\mathbf{A}$ matrix in Eq. (3.23), it is:

$$\Lambda = \{(0, 0)\}.$$

The current polynomial order is set to $k_{\text{curr}} = 1$.

Iteration 1

Polynomials from $\mathbf{A}$ (namely $\boldsymbol{\psi}_0$) are initially copied to $\mathbf{S}$. This is additionally completed by all multivariate polynomials of order $k_{\text{curr}} \leq 1$ (namely $\boldsymbol{\psi}_1$ and $\boldsymbol{\psi}_2$):

$$\mathbf{S} = \begin{bmatrix} \psi_0(\mathbf{z}_0) & \psi_1(\mathbf{z}_0) & \psi_2(\mathbf{z}_0) \\ \psi_0(\mathbf{z}_1) & \psi_1(\mathbf{z}_1) & \psi_2(\mathbf{z}_1) \\ \psi_0(\mathbf{z}_2) & \psi_1(\mathbf{z}_2) & \psi_2(\mathbf{z}_2) \\ \psi_0(\mathbf{z}_3) & \psi_1(\mathbf{z}_3) & \psi_2(\mathbf{z}_3) \\ \psi_0(\mathbf{z}_4) & \psi_1(\mathbf{z}_4) & \psi_2(\mathbf{z}_4) \\ \psi_0(\mathbf{z}_5) & \psi_1(\mathbf{z}_5) & \psi_2(\mathbf{z}_5) \end{bmatrix}.$$

According to Table 3.3, the polynomial basis Λ^+ for $k_{\text{curr}} = 1$ (which corresponds to the newly added polynomials $\boldsymbol{\psi}_{j_{\text{new}}} = \{\boldsymbol{\psi}_1, \boldsymbol{\psi}_2\}$) includes the following:

$$\Lambda^+ = \{(0,1),\ (1,0)\}.$$

An admissibility check is then performed, as described previously. As also shown in Figure 3.4, the current polynomial basis is already downward closed, so no modifications are necessary:

$$\Lambda^+_{\text{adm}} = \{(0,1),\ (1,0)\}.$$

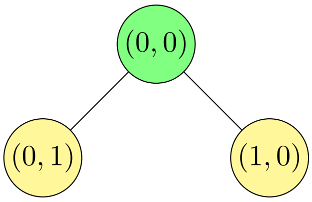

Figure 3.4: *Illustration of the adaptive multi-index selection process starting from the initial set $\Lambda = \{(0,0)\}$. The candidate multi-indices $\Lambda^+ = \{(0,1),(1,0)\}$ are highlighted in yellow. Since the polynomial basis is already downward closed, both candidates are admissible: $\Lambda^+_{adm} = \{(0,1),(1,0)\}$.*

The extended multi-index set becomes:

$$\Lambda_{\text{ext}} = \Lambda \cup \Lambda^+_{\text{adm}} = \{(0,0),\ (0,1),\ (1,0)\}.$$

More specifically, the corresponding newly admissible basis functions are:

$$\boldsymbol{\psi}_{j_{\text{new/adm}}} = \{\boldsymbol{\psi}_1, \boldsymbol{\psi}_2\}.$$

Therefore, $\mathbf{S}$ remains unchanged from its previously shown form containing $\boldsymbol{\psi}_0, \boldsymbol{\psi}_1$ and $\boldsymbol{\psi}_2$.

The number of available samples, $N = 6$, is then checked against the condition Eq. (3.17) $2P_{\mathbf{S}} \leq N$, where $P_{\mathbf{S}} = 3$ denotes the number of polynomial terms used in

$\mathbf{S}$. The condition is satisfied, so, no more samples are added at this point.

The $\mathbf{S}$ matrix, filled with the polynomials $\boldsymbol{\psi}_0, \boldsymbol{\psi}_1, \boldsymbol{\psi}_2$ at each sample point $\mathbf{z}_i$, is:

$$\mathbf{S} = \begin{bmatrix} \psi_0(\mathbf{z}_0) & \psi_1(\mathbf{z}_0) & \psi_2(\mathbf{z}_0) \\ \psi_0(\mathbf{z}_1) & \psi_1(\mathbf{z}_1) & \psi_2(\mathbf{z}_1) \\ \psi_0(\mathbf{z}_2) & \psi_1(\mathbf{z}_2) & \psi_2(\mathbf{z}_2) \\ \psi_0(\mathbf{z}_3) & \psi_1(\mathbf{z}_3) & \psi_2(\mathbf{z}_3) \\ \psi_0(\mathbf{z}_4) & \psi_1(\mathbf{z}_4) & \psi_2(\mathbf{z}_4) \\ \psi_0(\mathbf{z}_5) & \psi_1(\mathbf{z}_5) & \psi_2(\mathbf{z}_5) \end{bmatrix} = \begin{bmatrix} 1.0000 & 0.4803 & 1.1679 \\ 1.0000 & 0.4344 & -1.1579 \\ 1.0000 & -1.3101 & -1.1050 \\ 1.0000 & -0.1619 & 0.3224 \\ 1.0000 & 0.3828 & 1.0111 \\ 1.0000 & 0.7266 & 0.7779 \end{bmatrix}. \tag{3.24}$$

The problem is formulated as in Eq. (3.19), where $\boldsymbol{\alpha} = \begin{bmatrix}\alpha_0 & \alpha_1 & \alpha_2\end{bmatrix}^\top$ denotes the vector of PCE coefficients, and $\boldsymbol{f}$ is the vector of model evaluations at the sample points $\mathbf{x}_i$. Solving this system using OLS computes:

$$\alpha_0 = 1.0000000000, \quad \alpha_{1_{\text{new/adm}}} = 0.9999999999, \quad \alpha_{2_{\text{new/adm}}} = 2.0000000000.$$

The sensitivity indices corresponding to the new coefficients $\alpha_{j_{\text{new/adm}}}$ for $j = 1, 2$ are calculated, according to Eq. (3.20), and summarized in Table 3.7:

Coefficient $\alpha_{j_{\textbf{new/adm}}}$	**Sensitivity Index** $\eta_{j_{\textbf{new/adm}}}$	**Corresponding Polynomial**
$\alpha_{1_{\text{new/adm}}} = 0.9999$	0.9999	$\boldsymbol{\psi}_1$
$\alpha_{2_{\text{new/adm}}} = 2.0000$	4.0000	$\boldsymbol{\psi}_2$

Table 3.7: *Coefficients $\alpha_{j_{new/adm}}$ obtained from the latest solution of Eq.*(3.7)*, along with their sensitivity indices.*

$$\eta_{j_{\text{new/adm}}} > \frac{\eta_{\text{mean}}}{2} = 1.2499.$$

This inequality holds for the coefficient corresponding to $\boldsymbol{\psi}_2$, as shown in Table 3.7. Therefore, coefficient 0.9999 (corresponding to $\boldsymbol{\psi}_1$) is rejected for the time. Matrix $\mathbf{S}$ is emptied.

Consequently, the active matrix $\mathbf{A}$ retains only the basis functions associated with the significant term $\boldsymbol{\psi}_2$, along with the previously included $\boldsymbol{\psi}_0$.

$$\mathbf{A} = \begin{bmatrix} \psi_0(\mathbf{z}_0) & \psi_2(\mathbf{z}_0) \\ \psi_0(\mathbf{z}_1) & \psi_2(\mathbf{z}_1) \\ \psi_0(\mathbf{z}_2) & \psi_2(\mathbf{z}_2) \\ \psi_0(\mathbf{z}_3) & \psi_2(\mathbf{z}_3) \\ \psi_0(\mathbf{z}_4) & \psi_2(\mathbf{z}_4) \\ \psi_0(\mathbf{z}_5) & \psi_2(\mathbf{z}_5) \end{bmatrix} = \begin{bmatrix} 1.0000 & 1.1679 \\ 1.0000 & -1.1579 \\ 1.0000 & -1.1050 \\ 1.0000 & 0.3224 \\ 1.0000 & 1.0111 \\ 1.0000 & 0.7779 \end{bmatrix}. \tag{3.25}$$

In order to calculate the coefficients at this point, without being part of the algorithm (as this is done after all the iterations are completed), Eq.(3.22) is solved and the estimated coefficients are:

$$\alpha_0 = 1.0205941411, \quad \alpha_1 = 2.4215668481.$$

In this example, α_1 corresponds to the polynomial $\boldsymbol{\psi}_2$, and the value of α_0 has been updated. For completeness, $\varepsilon_{\mathrm{LOO}}$ is computed using Eq. (2.28), and is found to be

$$\varepsilon_{\mathrm{LOO}} = 0.1794.$$

Just for demonstration purposes, the surrogate model in this iteration is computed as:

$$\begin{aligned} \hat{F}(\mathbf{x}) &= \alpha_0\psi_0(\mathbf{z}) + \alpha_1\psi_2(\mathbf{z}) \\ &= 1.0205941411\frac{1}{\sqrt{0!}} + 2.4215668481\frac{\frac{x_1-\mu_1}{\sigma_1}}{\sqrt{1!}} \\ &= -0.19018928295 + 1.21078342405x_1. \end{aligned}$$

Figure 3.5 presents a comparison between the original function in Eq. (3.9) and its surrogate approximation $\hat{F}(\mathbf{x}) = -0.1902 + 1.2108\,x_1$ over the domain $[-3,3]^2$. The darker surface corresponds to the original function, while the lighter surface depicts the surrogate, illustrating their respective shapes in the 3D space.

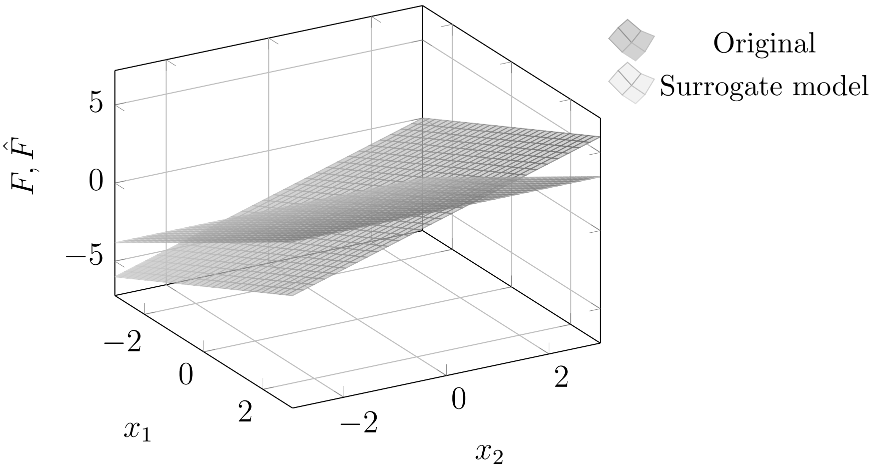

Figure 3.5: *Comparison between the original function in Eq.* (3.9) *(darker surface) and the surrogate model* $\hat{F}(\mathbf{x}) = -0.1902 + 1.2108\,x_1$ *(lighter surface) over the domain* $x_1, x_2 \in [-3, 3]$.

The polynomial order is increased to $k_{\text{curr}} = 2$. Since no termination criterion is activated, the algorithm proceeds to iteration 2.

Iteration 2

Since all basis functions currently included in the $\mathbf{A}$ matrix are considered significant—namely $\boldsymbol{\psi}_0$ and $\boldsymbol{\psi}_2$—and based on their univariate orders as shown in Table 3.3, it follows that $\Lambda = \{(0,0),\ (1,0)\}$.

In this iteration, the polynomial basis Λ^+ is examined, considering all polynomials of order $k = 0, 1$, and 2, i.e., up to and including the current chaos order $k_{\text{curr}} = 2$. This includes the basis functions $\boldsymbol{\psi}_3$, $\boldsymbol{\psi}_4$, $\boldsymbol{\psi}_5$, and $\boldsymbol{\psi}_1$, as the remaining basis functions are already included in $\mathbf{A}$.

Thus, the candidate multi-indices to be examined first—according to the downward-closed criterion—are:

$$\Lambda^+ = \{(0,2),\ (1,1),\ (2,0),\ (0,1)\}.$$

An admissibility check is then performed, as described previously. As also shown in Figure 3.6, indices $(0,2)$ and $(1,1)$ are rejected due to violation of the downward closedness condition. Therefore, the updated admissible polynomial basis becomes:

$$\Lambda^+_{\text{adm}} = \{(2,0),\ (0,1)\}.$$

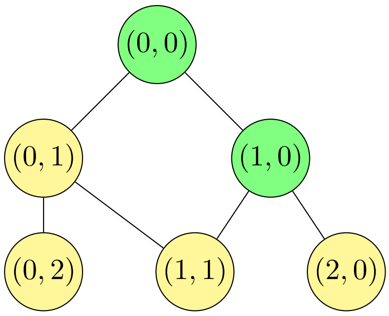

Figure 3.6: *The root node* $(0,0)$ *and its immediate successor* $(1,0)$ *are already in the basis (green). Candidate multi-indices (yellow) include* $(0,1)$*,* $(0,2)$*,* $(1,1)$*, and* $(2,0)$*. Nodes* $(1,1)$ *and* $(0,2)$ *are rejected due to missing predecessors in the index set, while* $(0,1)$ *and* $(2,0)$ *are accepted and added to the admissible set, updating the selection.*

As a result, the extended multi-index set becomes:

$$\Lambda_{\text{ext}} = \Lambda \cup \Lambda^{+}_{\text{adm}} = \{(0,0),\ (1,0),\ (2,0),\ (0,1)\}\,.$$

Looking at Table 3.3, the basis functions of Λ_{ext} that will be added to the $\mathbf{S}$ matrix are $\boldsymbol{\psi}_j = \{\boldsymbol{\psi}_0, \boldsymbol{\psi}_2, \boldsymbol{\psi}_5, \boldsymbol{\psi}_1\}$, with the newly admitted basis functions $\boldsymbol{\psi}_{j_{\text{new/adm}}} = \{\boldsymbol{\psi}_5, \boldsymbol{\psi}_1\}$.

The basis is intentionally kept in this order—rather than being sorted by increasing total degree—to facilitate computational extraction of the newly added basis functions $(\boldsymbol{\psi}_5, \boldsymbol{\psi}_1)$ if needed. Since they appear at the tail of the list, there is no need to define a separate index vector to identify their positions within the basis.

The number of available samples, $N = 6$, is then checked against the condition in Eq. (3.17), $2P_{\mathbf{S}} \leq N$, where $P_{\mathbf{S}} = 4$. The condition is not satisfied, so according to Eq. (3.18), 2 additional samples are needed. The updated collection of input samples $\mathbf{z}_i$ and their corresponding evaluations $F(x_{i,1}, x_{i,2})$ is shown in Table 3.8.

The design matrix $\mathbf{S}$ is constructed by calculating the polynomial basis functions $\boldsymbol{\psi}_0, \boldsymbol{\psi}_2, \boldsymbol{\psi}_5\ \boldsymbol{\psi}_1$ at each sample point $\mathbf{z}_i$:

$$\mathbf{S} = \begin{bmatrix} \psi_0(\mathbf{z}_0) & \psi_2(\mathbf{z}_0) & \psi_5(\mathbf{z}_0) & \psi_1(\mathbf{z}_0) \\ \psi_0(\mathbf{z}_1) & \psi_2(\mathbf{z}_1) & \psi_5(\mathbf{z}_1) & \psi_1(\mathbf{z}_1) \\ \vdots & \vdots & \vdots & \vdots \\ \psi_0(\mathbf{z}_7) & \psi_2(\mathbf{z}_7) & \psi_5(\mathbf{z}_7) & \psi_1(\mathbf{z}_7) \end{bmatrix} = \begin{bmatrix} 1.0000 & 1.1679 & 0.2574 & 0.4803 \\ 1.0000 & -1.1579 & 0.2409 & 0.4344 \\ 1.0000 & -1.1050 & 0.1563 & -1.3101 \\ 1.0000 & 0.3224 & -0.6336 & -0.1619 \\ 1.0000 & 1.0111 & 0.0158 & 0.3828 \\ 1.0000 & 0.7779 & -0.2793 & 0.7266 \\ 1.0000 & -1.2157 & 0.3379 & -0.0938 \\ 1.0000 & 1.0665 & 0.0972 & 0.5372 \end{bmatrix}. \tag{3.26}$$

Index i	Input Samples $\mathbf{z}_i = (z_{i,1}, z_{i,2})$		$\mathbf{x}_i = (x_{i,1}, x_{i,2})$		Output $F(x_{i,1}, x_{i,2})$
0	1.1679	0.4803	3.3358	0.4803	3.8161 (known)
1	-1.1579	0.4344	-1.3158	0.4344	-0.8814 (known)
2	-1.1050	-1.3101	-1.2100	-1.3101	-2.5201(known)
3	0.3224	-0.1619	1.6449	-0.1619	1.4830 (known)
4	1.0111	0.3828	3.0222	0.3828	3.4050 (known)
5	0.7779	0.7266	2.5557	0.7266	3.2824 (known)
6	-1.2157	-0.0938	-1.4314	-0.0938	-1.5252
7	1.0665	0.5372	3.1330	0.5372	3.6702

Table 3.8: *Standardized input samples $\mathbf{z}_i \sim \mathcal{N}(0,1)^2$, corresponding transformed values $\mathbf{x}_i$, and the output values $F(x_{i,1}, x_{i,2})$ used for regression. Outputs 0–5 are already evaluated and known from the previous iteration.*

Formulating the problem as in Eq. (3.19), where $\boldsymbol{\alpha} = \begin{bmatrix}\alpha_0 & \alpha_1 & \alpha_{2_{\text{new/adm}}} & \alpha_{3_{\text{new/adm}}}\end{bmatrix}^\top$ denotes the vector of PCE coefficients, and $\boldsymbol{f}$ is the vector of model evaluations at the sample points $\mathbf{x}_i$.

The solution of this system using OLS results:

$$\alpha_0 = 0.9999999999, \quad \alpha_1 = 1.9999999999,$$
$$\alpha_{2_{\text{new/adm}}} = 0.0000000000, \quad \alpha_{3_{\text{new/adm}}} = 1.000000000.$$

The contributions of the newly added basis functions, $\boldsymbol{\psi}_5$ and $\boldsymbol{\psi}_1$, according to Eq. (3.20), are reported in Table 3.9.

Coefficient $\alpha_{j_{\text{new/adm}}}$	Sensitivity Index $\eta_{j_{\text{new/adm}}}$	Corresponding Polynomial
$\alpha_{2_{\text{new/adm}}} = 0.0000$	0.0000	$\boldsymbol{\psi}_5$
$\alpha_{3_{\text{new/adm}}} = 1.0000$	1.0000	$\boldsymbol{\psi}_1$

Table 3.9: *Coefficients $\alpha_{2_{new/adm}}$ and $\alpha_{3_{new/adm}}$ obtained from the solution of Eq.(3.19), along with their sensitivity indices.*

The coefficient $\alpha_{3_{\text{new/adm}}} = 0.9999$ (corresponding to $\boldsymbol{\psi}_1$) is selected according to Eq. (3.21). Therefore, the active matrix $\mathbf{A}$ becomes:

$$\mathbf{A} = \begin{bmatrix} \psi_0(\mathbf{z}_0) & \psi_2(\mathbf{z}_0) & \psi_1(\mathbf{z}_0) \\ \psi_0(\mathbf{z}_1) & \psi_2(\mathbf{z}_1) & \psi_1(\mathbf{z}_1) \\ \vdots & \vdots & \vdots \\ \psi_0(\mathbf{z}_7) & \psi_2(\mathbf{z}_7) & \psi_1(\mathbf{z}_7) \end{bmatrix} = \begin{bmatrix} 1.0000 & 1.1679 & 0.4803 \\ 1.0000 & -1.1579 & 0.4344 \\ 1.0000 & -1.1050 & -1.3101 \\ 1.0000 & 0.3224 & -0.1619 \\ 1.0000 & 1.0111 & 0.3828 \\ 1.0000 & 0.7779 & 0.7266 \\ 1.0000 & -1.2157 & -0.0938 \\ 1.0000 & 1.0665 & 0.5372 \end{bmatrix}. \tag{3.27}$$

Since $k_{\text{curr}} = k_{\text{final}}$ has been reached, the iterations are terminated and the system in Eq. (3.22) is solved using OLS, which computes:

$$\alpha_0 = 1.0000000000, \quad \alpha_1 = 2.0000000000, \quad \alpha_2 = 1.0000000000.$$

The LOO error is found to be:

$$\varepsilon_{\text{LOO}} = 0.000000.$$

This result of the LOO error is expected, as the surrogate model exactly reproduces the original model, which was also confirmed earlier in OMP by the identical coefficients. Consequently, the mean and standard deviation remain the same as the analytical results.

The algorithm terminates once the step corresponding to $k_{\text{curr}} = k_{\text{final}}$ has been executed. The first two statistical moments, computed using the PCE coefficients from iteration 2, are:

$$\begin{aligned} \text{Mean:} \quad & \mu_{\hat{F}} = \alpha_0 = 1.0000000000, \\ \text{Standard Deviation:} \quad & \sigma_{\hat{F}} = \sqrt{\alpha_1^2 + \alpha_2^2} = 2.2360679775. \end{aligned}$$

which match the analytical solution derived above.

The two sparse UQ methods employed yielded the exact analytical expression. It is important to note that these values do not represent the computational efficiency of the methods in the general case, as the examples presented here are solely for explanatory purposes. Both methods are fundamentally designed to handle problems with a high number of uncertain variables, where their efficiency should be evaluated.

In the following, two numerical applications of the sparse methods OMP and ESCAPE are presented, for some first comparisons to be made.

3.2 Pseudo-Engineering Problems

This section presents a comperative analysis of OMP and ESCAPE to assess their accuracy and efficiency, with particular emphasis on the number of evaluations required. They are compared to each other using Monte-Carlo simulations, which serve as the reference solution ('ground truth'). The comparison is conducted across two distinct problem scenarios to evaluate the performance of each method under varying conditions, with the goal of identifying the optimal balance between computational cost and accuracy.

In addition to evaluating the statistical estimates, the analysis will also examine the polynomial bases constructed by the two methods, providing insight into which basis functions are selected by each algorithm and how each basis is adapted to the problem at hand. All regression algorithms in the following problems will operate with $k = 2$ (for ESCAPE, also $k_{\text{final}} = 2$).

3.2.1 Problem 1: Wing Weight

The wing weight function from [10], adapted from the aircraft design handbook [22], models the wing of a Cessna C172 Skyhawk. It is used for sensitivity analysis in aerospace and depends on factors such as wing area, fuel weight in the wing, aspect ratio, quarter-chord sweep angle, dynamic pressure at cruise, taper ratio, airfoil thickness-to-chord ratio, ultimate load factor, flight design gross weight, and paint weight.

The light aircraft wing weight (QoI) is defined as follows:

$$F(\mathbf{x}) = 0.036 S_w^{0.758} W_{fw}^{0.0035} \left(\frac{A}{\cos^2(\Lambda)}\right)^{0.6} q^{0.006} \ell^{0.04} \left(\frac{100 t_c}{\cos(\Lambda)}\right)^{-0.3} (N_z W_{dg})^{0.49} + S_w W_p. \tag{3.28}$$

The $D = 10$ uncertain inputs are assumed to follow normal distributions, in Table 3.10.

Inputs	Unit	Notation	Mean	Standard Deviation
Wing area	ft^2	S_w	175	14.43
Weight of fuel in wing	lb	W_{fw}	260	23.09
Aspect ratio	–	A	8	1.15
Quarter-chord sweep	deg	Λ	0	5.773
Dynamic pressure at cruise	$\mathrm{lb/ft}^2$	q	30.5	8.38
Taper ratio	–	ℓ	0.75	0.14
Aerofoil thickness/chord ratio	–	t_c	0.13	0.029
Ultimate load factor	–	N_z	4.25	1.01
Flight design gross weight	lb	W_{dg}	2100	231.08
Paint weight	$\mathrm{lb/ft}^2$	W_p	0.0525	0.0159

Table 3.10: *Problem 1: Normal distributions of stochastic inputs [22].*

Parametric study of the sample-multiplier factor m in ESCAPE method

A parametric study on the factor m of the ESCAPE method, which appears in Eq. (3.18) as part of Step 6, was conducted. The method was executed for various values of $m \in \{1.5, 2, 2.5, 3, 3.5, 4\}$, and the results are illustrated in Figure 3.7, which shows the evolution of the mean and standard deviation, as a function of the number of function evaluations (computational cost). As observed, the choice $m = 2$ offers the most favorable trade-off between accuracy and computational cost. Based on this outcome, this value of m is adopted throughout this thesis.

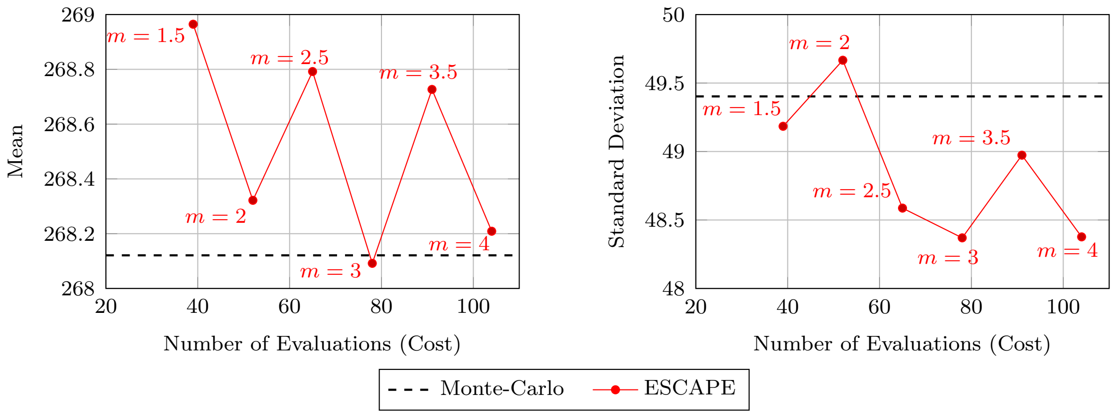

Figure 3.7: *Problem 1: Parametric study of the factor m in the ESCAPE method (Step 6). Left: Mean value of the QoI. Right: Corresponding standard deviation. Dashed lines indicate the converged Monte-Carlo reference values ('ground truth').*

The non-sparse basis using $k = 2$ according to Eq.(2.2) has $P_{\text{total}} = 66$ basis functions.

Parametric study of the sample-multiplier factor b in OMP method

A parametric study is conducted to investigate the effect of the factor b, as defined in Eq. (3.4), on the accuracy of the mean and standard deviation estimates. As shown in Figure 3.8, different values of b are tested, and the resulting values are compared against the Monte-Carlo 'ground truth' values. The results indicate that $b = 5$ achieves a good balance between accuracy and computational cost, justifying its selection in this thesis.

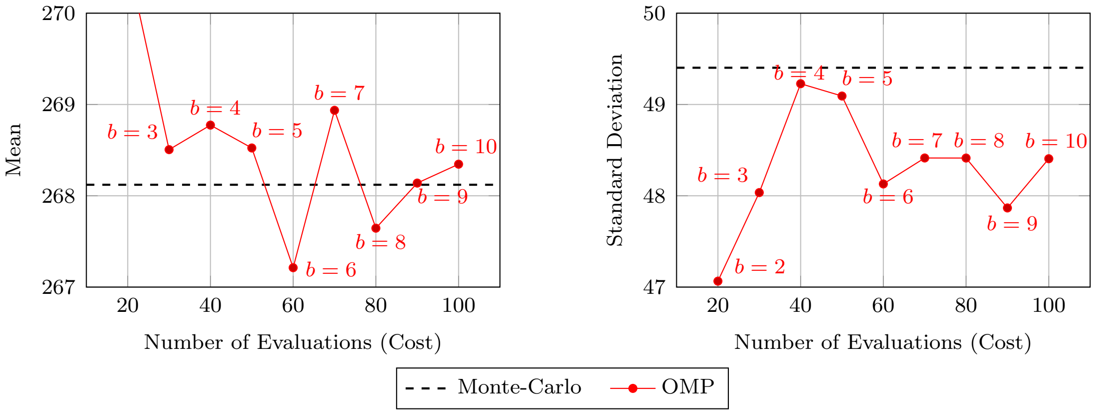

Figure 3.8: *Problem 1: Parametric study of the factor b in the OMP method. Left: Mean value of the QoI. Right: Corresponding standard deviation. Dashed lines indicate the converged Monte-Carlo reference values ('ground truth').*

The bar charts in Figures 3.10 and 3.9a show results obtained using NSPCE with $SR = 3$ and $N = 198$. Bar charts in Figure 3.9a illustrate the Sobol total sensitivity indices. This chart quantifies the relative importance of each stochastic input to the QoI. The right chart (b) shows the total number of non-zeroth-order univariate polynomial terms selected per input variable by the OMP (blue) and ESCAPE (red) algorithms. More precisely, it presents the number of non-zero univariate terms retained in the final sparse polynomial basis for each stochastic input.

For example, consider the following final, sparse polynomial basis $\mathbf{A}$ with two stochastic variables, constructed as tensor products of univariate basis functions p_0, p_1, p_2, in variables x_1 and x_2:

$$\mathbf{A} = \begin{bmatrix} p_0(z_{0,1})\, p_0(z_{0,2}) & p_0(z_{0,1})\, p_1(z_{0,2}) & p_0(z_{0,1})\, p_2(z_{0,2}) \\ p_0(z_{1,1})\, p_0(z_{1,2}) & p_0(z_{1,1})\, p_1(z_{1,2}) & p_0(z_{1,1})\, p_2(z_{1,2}) \\ \vdots & \vdots & \vdots \\ p_0(z_{N-1,1})\, p_0(z_{N-1,2}) & p_0(z_{N-1,1})\, p_1(z_{N-1,2}) & p_0(z_{N-1,1})\, p_2(z_{N-1,2}) \end{bmatrix}.$$

In this example, the polynomial basis in x_1 contains only the zeroth-order term, while those in x_2 range from order 0 to 2, resulting in:

- Number of non-zero orders in x_1: 0
- Number of non-zero orders in x_2: 2 (orders 1 and 2)

Figure 3.9b is directly compared to the reference Sobol indices in Figure 3.9a, in order to assess how effectively each adaptive method identifies and emphasizes the most influential variables for the variance of the QoI. The comparison illustrates how closely the selected polynomial structures reflect the true sensitivity ranking of the input variables, thus indicating the quality of each method's basis construction.

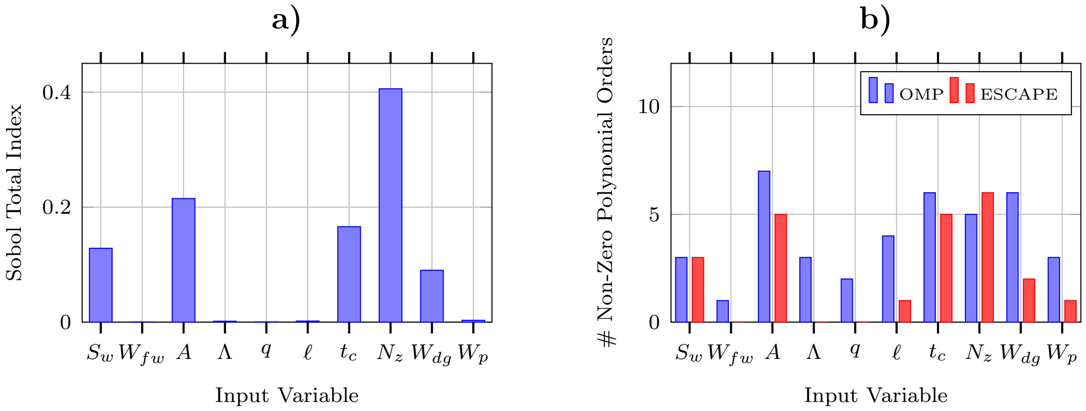

Figure 3.9: *Problem 1: (a) Sobol total sensitivity indices (to be used as reference) derived using NSPCE, quantifying each input variable's contribution to output variance. (b) an adaptivity-based comparison between OMP (blue) and ESCAPE (red), showing the count of non-zeroth-order univariate polynomial terms per input dimension in the constructed polynomial basis. These counts indicate the degree of importance each algorithm assigns to the input variables and are compared to subfigure (a).*

Figure 3.9a shows that some uncertain inputs, such as W_{fw}, Λ, q, ℓ, and W_p, are not significantly influential for the variance of the QoI, according to their Sobol indices. Figure 3.9b demonstrates that both OMP and ESCAPE correctly assigns more non-zeroth-order polynomial terms to the inputs S_w, A, t_c, W_{dg}, and N_z, reflecting their higher influence on the QoI. However, the ESCAPE method appears to construct its polynomial basis more cautiously and in closer agreement with the significance pattern indicated by the Sobol analysis, suggesting a more targeted and informed adaptivity than OMP. It is worth noting that the input ℓ may be overestimated in the basis constructed by ESCAPE. The computational cost for each method shown in Figure 3.9b is reported in Table 3.11.

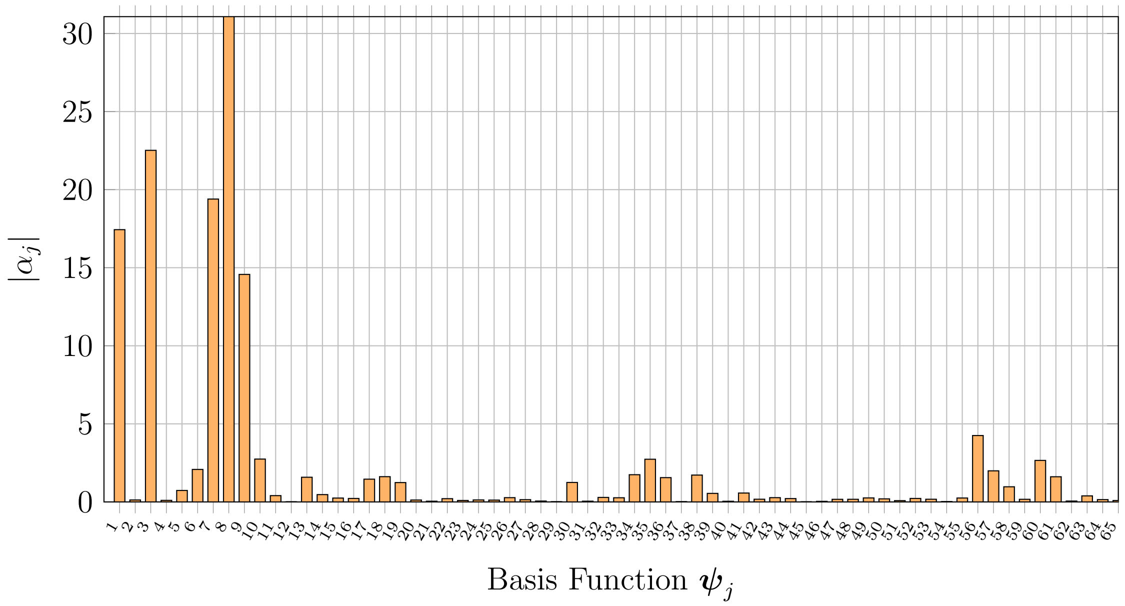

Figure 3.10: *Problem 1: Bar chart of the absolute values of the PCE coefficients with the corresponding polynomial basis functions $\boldsymbol{\psi}_j$ on the x-axis, excluding the constant term $\boldsymbol{\psi}_0$. The coefficients are obtained using NSPCE.*

Another interesting comparison, shown in Figure 3.10, presents the absolute values of the PCE coefficients associated with the non-sparse polynomial basis functions $\boldsymbol{\psi}_j$, excluding the constant term $\boldsymbol{\psi}_0$. The bar chart highlights the relative importance of different basis functions in the expansion. A few coefficients, such as those corresponding to $\boldsymbol{\psi}_3$, $\boldsymbol{\psi}_8$, $\boldsymbol{\psi}_7$, $\boldsymbol{\psi}_9$, and $\boldsymbol{\psi}_{56}$, dominate with significantly larger magnitudes, while most others remain small. This, indicates that only a limited subset of basis functions contributes substantially to the system's response. In order to compare whether OMP and ESCAPE correctly select the important basis functions and reject the rest, a sorted version of these bars is presented in Figure 3.11. Here, the blue dots represent the subset of basis functions selected by the OMP, while the red triangles denote those identified by the ESCAPE method. This comparison highlights the capability of sparse techniques to add the most influential basis functions to the sparse basis, while filtering out less relevant ones. In particular, both methods identify, among others, the ten most relevant basis functions, namely those corresponding to $j = \{8, 3, 7, 1, 9, 56, 10, 35, 60, 6\}$. They also appear to skip some not irrelevant basis functions, such as $j = 57, 61, 13$ before choosing others of lesser importance. Beyond that, OMP tends to include a few less significant basis functions, like $j = 64$ and 62, which may not be optimal since it unnecessarily increases the size of the basis.

It is worth noting that $\varepsilon_{\mathrm{LOO}}$ remains below 0.01, thereby confirming the reliability of the results. As shown in Table 3.11, the reported information includes the CR, as defined in Eq. (1.3), P_{sparse}, along with estimates of the mean $\mu_{\hat{F}}$ and standard deviation $\sigma_{\hat{F}}$ of the QoI. The estimates from both OMP and ESCAPE, which require

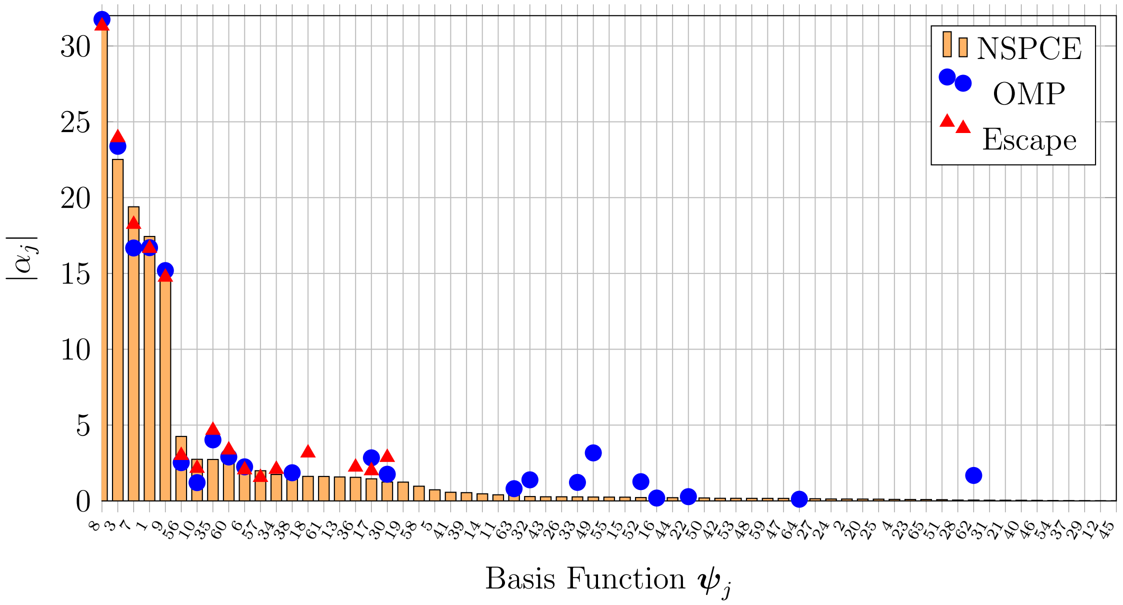

Figure 3.11: *Problem 1: Bar chart of the sorted, absolute values of the PCE coefficients, excluding the constant basis function $\boldsymbol{\psi}_0$, together with the polynomials and their coefficient magnitudes selected by OMP (blue dots) and ESCAPE (red triangles).*

Method (k)	$\mu_{\hat{F}}$	$\sigma_{\hat{F}}$	P_{sparse}/CR	Cost (Function Calls)
Monte-Carlo	**268.120836175**	**49.402464061**	–	$\mathbf{10^9}$
OMP (2)	268.5235342628	49.0926092981	22/0.3	50
ESCAPE (2)	268.3217099425	49.6664607313	17/0.26	52

Table 3.11: *Problem 1: Comparison of the OMP and ESCAPE methods for estimating the mean $\mu_{\hat{F}}$ and standard deviation $\sigma_{\hat{F}}$ of the QoI, also compared with Monte-Carlo. The table also reports P_{sparse}, CR and the number of function evaluations.*

nearly the same computational cost, seem very close to the Monte-Carlo values.

3.2.2 Problem 2: Beam's deflection

A simply supported beam under uniform load is considered, with artificially increased input dimensionality, up to $D = 20$, to examine its impact on the PCE methods. The beam's deflection at coordinate ℓ_m is given by

$$\delta(\ell_m) = \frac{P\ell_m \left(L^3 - 2\ell_m^2 L + \ell_m^3\right)}{2Ewh^3} + 10^{-10} \sum_{i=6}^{20} d_i, \tag{3.29}$$

where $\ell_m = \frac{mL}{M+1}, \ m = 1, \ldots, M$. The five primary input parameters are assumed to follow normal distributions, as specified below. Additionally, 15 dummy parameters $d_i, \ i = 6, \ldots, 20$ are introduced, also normally distributed, though they do not influence the response. Despite their irrelevance, the increased input dimensionality affects the performance of the PCE methods. This setup allows us to evaluate whether the methods can handle situations where the curse of dimensionality becomes significant [14].

The non-sparse basis for order $k = 2$, computed according to Eq. (2.2), consists of a total of $P_{\text{total}} = 231$ basis functions.

Inputs	Unit	Notation	Mean	Standard Deviation
Width	m	w	$15 \cdot 10^{-2}$	$75 \cdot 10^{-4}$
Height	m	h	$3 \cdot 10^{-1}$	$15 \cdot 10^{-3}$
Length	m	L	5	$5 \cdot 10^{-2}$
Young's modulus	Pa	E	$3 \cdot 10^{10}$	$45 \cdot 10^{8}$
Load	N/m	P	$1 \cdot 10^{4}$	$2 \cdot 10^{3}$
Dummy	–	$d_6 - d_{20}$	1	1

Table 3.12: *Problem 2: Normal distributions of stochastic inputs [14].*

Figure 3.12 shows (a) Sobol total sensitivity indices, obtained via NSPCE, and (b) the number of non-zero univariate polynomial terms per stochastic input in the polynomial bases constructed by OMP (blue) and ESCAPE (red). The latter reflects the adaptivity of each algorithm in relation to the Sobol reference ('ground truth'). As a result, in contrast to OMP, ESCAPE effectively ignores the purposely insignificant orders corresponding to d_6 through d_{20} for the QoI, which is useful, as it significantly reduces the size of the polynomial basis. Additionally, both methods identified that h, E, P inputs have increased importance in the variance of the QoI, according to the Sobol indices shown in Figure 3.12a, potentially enabling the construction of more appropriate polynomial basis.

In order to evaluate whether OMP and ESCAPE correctly identify the dominant basis functions while discarding the less relevant ones, Figure 3.13 displays the sorted absolute values of the PCE coefficients. The blue dots represent the subset of basis functions selected by OMP, while the red triangles correspond to those identified by ESCAPE. Both methods capture several of the most influential basis functions (e.g., $j = \{5, 4, 2, 1, 3, 44, 43\}$), thereby demonstrating their capability to construct a compact sparse basis. However, ESCAPE appears to recover a larger fraction of the important basis functions (such as $j = 41, 61, 62$), while both methods miss others like $j = 79$. They both seem to select also less relevant terms, which are omitted from the figure due to space limitations.

Table 3.13 presents the mean and standard deviation estimates obtained by the OMP and ESCAPE methods, alongside the converged Monte-Carlo reference results ('ground truth'). Notably, ε_{LOO} has remained below 0.01, thereby confirming the

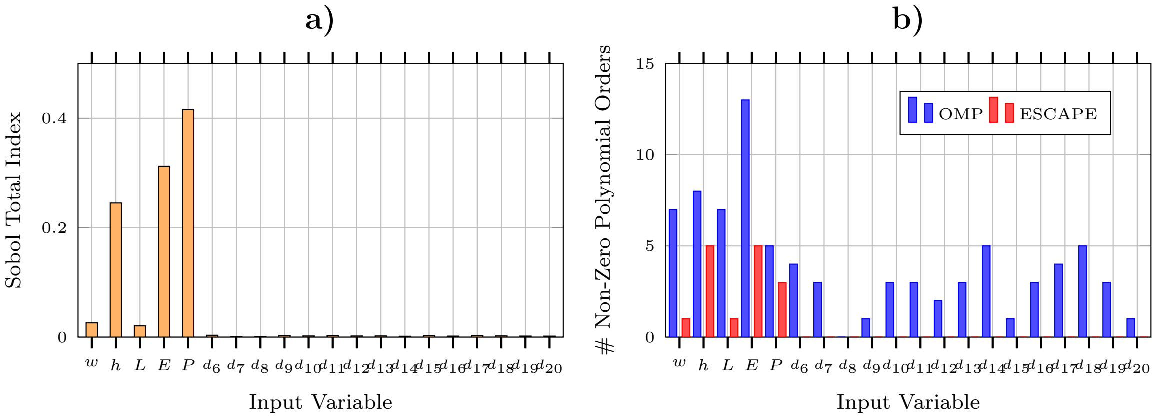

Figure 3.12: *Problem 2: (a) Sobol total sensitivity indices obtained with NSPCE (used as reference), quantifying the contribution of each input variable to the output variance. (b) Adaptivity-based comparison between OMP (blue) and ESCAPE (red), showing the number of non-zeroth-order univariate polynomial terms per input dimension in the constructed basis. These counts reflect the importance each algorithm assigns to the stochastic variables and are contrasted with subfigure (a).*

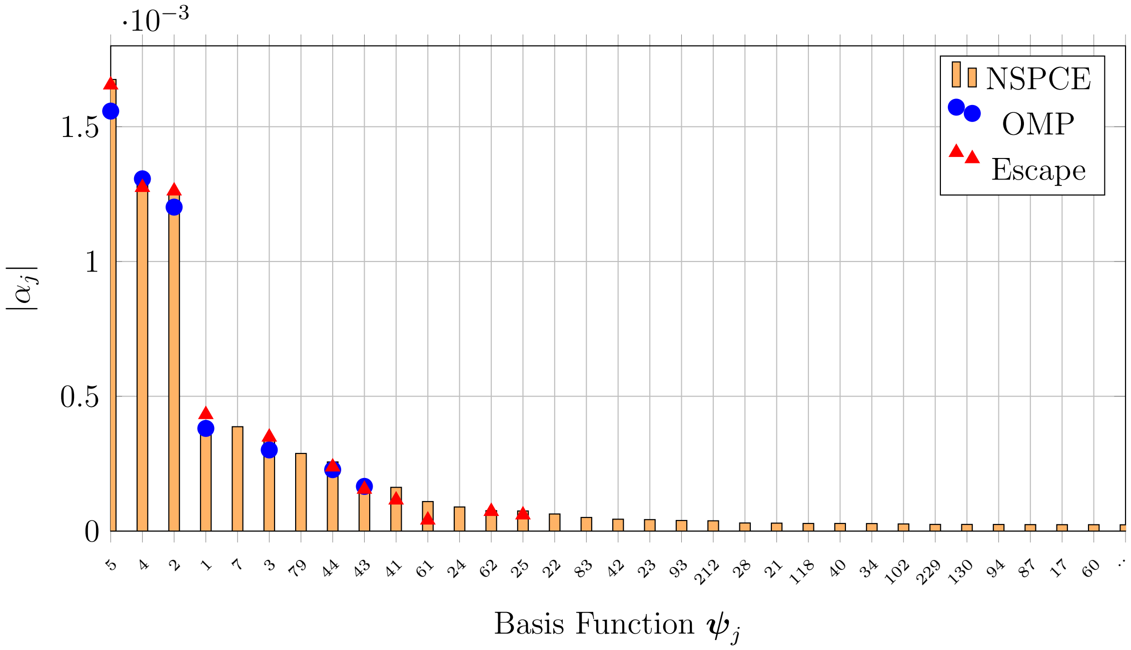

Figure 3.13: *Problem 2: Bar chart of the sorted absolute values of the first subset of PCE coefficients (out of the total $P_{total} = 231$), shown for clarity, excluding the constant basis function ψ_0. The markers indicate the coefficients selected by OMP (blue dots) and ESCAPE (red triangles).*

reliability of the results. OMP demonstrates a good balance between computational cost and accuracy, while ESCAPE achieves a similarly accurate representation using a sparser polynomial basis and less function evaluations.

Method (k)	$\mu_{\hat{F}}$	$\sigma_{\hat{F}}$	P_{sparse}/CR	Cost (Function Calls)
Monte-Carlo	**0.0083836936**	**0.0026087982**	–	$\mathbf{10^7}$
OMP (2)	0.0083872707	0.0025527939	48/0.21	100
ESCAPE (2)	0.0083750489	0.0025528164	13/0.06	72

Table 3.13: *Problem 2: Comparison of the OMP and ESCAPE methods in estimating the mean $\mu_{\hat{F}}$ and standard deviation $\sigma_{\hat{F}}$ of the QoI, alongside Monte-Carlo results. The table additionally presents P_{sparse}, CR, and the number of function evaluations.*

3.3 Conclusions

For Problem 1, ESCAPE achieved a compression ratio of 0.26 compared to 0.30 for OMP, indicating the construction of a smaller polynomial basis. Both methods reproduced the mean and standard deviation of the Monte Carlo reference with errors below 0.7%, with ESCAPE yielding slightly higher accuracy. The number of function evaluations was nearly identical, with 52 for ESCAPE and 50 for OMP.

For Problem 2, ESCAPE achieved again a substantially lower CR of 0.06 compared to 0.21 for OMP. Accuracy with respect to the Monte Carlo reference remained within 2%, with OMP providing slightly better prediction of the mean. It is worth noting that ESCAPE required fewer function evaluations (72) than OMP (100).

Overall, these results show that both ESCAPE and OMP achieve similarly high accuracy in estimating the mean $\mu_{\hat{F}}$ and standard deviation $\sigma_{\hat{F}}$ of the QoI.

Chapter 4

UQ in Aerodynamic Applications

In this chapter, OMP and ESCAPE methods are applied to three aerodynamic case studies that involve normally distributed, geometric (using shape deformation techniques) and flow uncertainties, such as inlet velocity.

For shape parameterization, B-spline-based Free-Form Deformation (FFD) is employed. In particular, volumetric B-splines are used to map all CFD mesh points located within predefined morphing boxes from the Cartesian space (x, y) to a parametric space (u, v). This mapping is performed once at the beginning of each case.

Each case involves the solution of the Navier–Stokes equations using the OpenFOAM software. For the present study, no turbulence model was employed, and the simulations were conducted under laminar assumptions. The aim is to assess the reliability and effectiveness of the proposed sparse UQ methods in both internal and external aerodynamics. This is accomplished by comparing the predicted means and standard deviations with those obtained via the Monte-Carlo method, thereby demonstrating the accuracy and practical applicability of the sparse techniques in real-world aerodynamic problems.

For the first case, however, Monte-Carlo simulations are prohibitively expensive to run to convergence. Therefore, NSPCE with $k = 3$ is used as the reference solution instead.

4.1 Problem 3: NACA0012 Airfoil

The NACA0012 airfoil is selected for this study due to its symmetric design, which ideally produces zero lift under parallel flow conditions. Small geometric perturbations—namely, 8 uncertainty inputs corresponding to the coordinates of Control

Points (CPs), on the order of 0.01% of the chord length are considered in this example. These perturbations can influence the aerodynamic performance, such as the lift coefficient, which serves as the QoI in this case. The effects are quantified in the following sections and used to compare the performance of ESCAPE and OMP. According to Eq. (2.2), for a total polynomial order $k = 2$, the total number of non-sparse polynomials is $P_{\text{total}} = 45$.

Model Description and Flow Conditions

This study investigates the external flow around an isolated NACA 0012 airfoil at a freestream flow angle of 0° under shape/geometrical uncertainties. The airfoil has a chord length $C = 1$ m and a maximum thickness equal to 12% of the chord, i.e., $0.12\,C$. The computational domain is discretized using a structured mesh comprising 37,800 hexahedral cells. The corresponding mesh is illustrated in Figure 4.1.

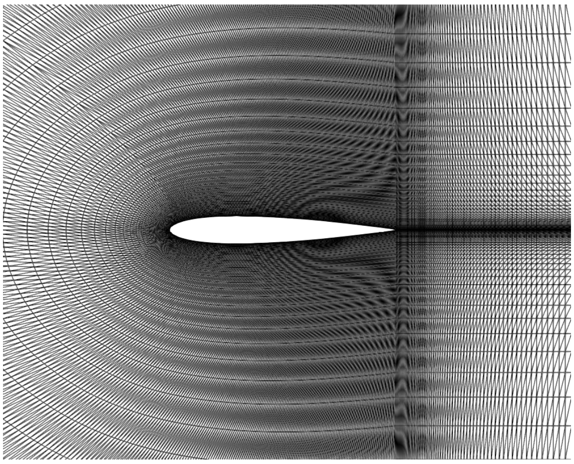

Figure 4.1: *Problem 3: Computational mesh.*

Regarding the laminar flow conditions, air flows at a velocity of 6 m/s. The flow is assumed incompressible, and the working fluid is modeled as Newtonian with a kinematic viscosity of $\nu = 6 \times 10^{-3}\,\text{m}^2/\text{s}$.

The no-slip boundary condition is imposed along the airfoil walls. For the pressure field, a zero-gradient boundary condition is imposed at the airfoil surfaces.

The aim of this application is to perform UQ with respect to a QoI, which is defined as the aerodynamic lift coefficient C_L. This coefficient is computed by projecting the total aerodynamic force F onto a given lift direction $\mathbf{e}_L$, and normalizing it with the dynamic pressure and a reference area. The mathematical expression is given by

$$J = C_L = \frac{F \cdot \mathbf{e}_L}{\frac{1}{2}\rho_\infty U_\infty^2 A_{\text{ref}}},$$

where ρ_∞ denotes the freestream density, U_∞ the freestream velocity magnitude, and A_{ref} the chosen reference area.

In Figure 4.2, the CPs that parameterize the geometry of the airfoil using the B-splines method are shown. Table 4.1 presents the mean values and standard deviations of the normal distributions assigned to the displacements of these CPs. The blue CPs are fixed: specifically, the first column remains stationary to prevent mesh distortion, and the second column is inactive to preserve continuity in slope. The red CPs are active and can be displaced during the UQ process to represent geometrical uncertainty. Along the boundaries of the morphing box, two series of control points remain still, and this ensures a smooth transition between the controlled and uncontrolled parts of the CFD grid.

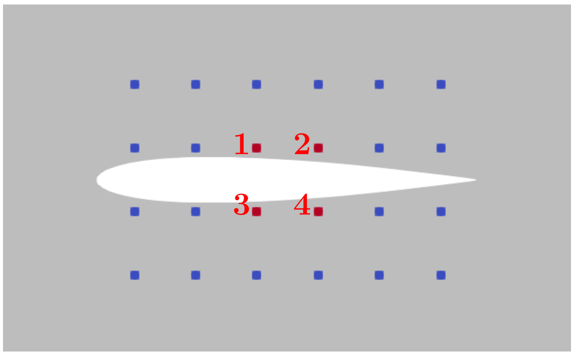

Figure 4.2: *Problem 3: Morphing box of the NACA0012 airfoil.*

Uncertain Parameters:	Mean (μ)	Standard Dev. (σ)
Displacements $\Delta x, \Delta y$ of CPs 1–4 (8 inputs)	$0\,C$	$0.0001\,C$

Table 4.1: *Problem 3: Mean and standard deviation of the normal distributions used for the displacement of CPs 1–4, expressed in terms of the chord length C.*

NSPCE with $k = 3$ and 165 polynomials is used to provide a reliable result for validation of the 2 sparse methods. The convergence of the first two statistical moments—mean and standard deviation using $D = 8$ uncertain variables is illustrated in Figure 4.3. These results serve to assess the accuracy of the sparse UQ methods.

In Figure 4.4, the convergence of the two algorithms, ESCAPE and OMP, can be observed. Setting $k_{final} = 3$ for the ESCAPE method, as described in Chapter 3, implies that the algorithm performs three iterations. In each iteration $i \in \{1, 2, 3\}$, the algorithm may select significant polynomial terms of order $k \leq i$. For example, during iteration 3, polynomial terms of order up to 3 (i.e. orders 1, 2, and 3) can be included in the model, provided they are found to be significant.

In Figure 4.5, the ε_{LOO} errors for both methods are shown. Both methods achieve low and acceptable values of ε_{LOO}. A direct comparison between the values for each method is not strictly meaningful, since ε_{LOO} depends on the specific system solved in each case.

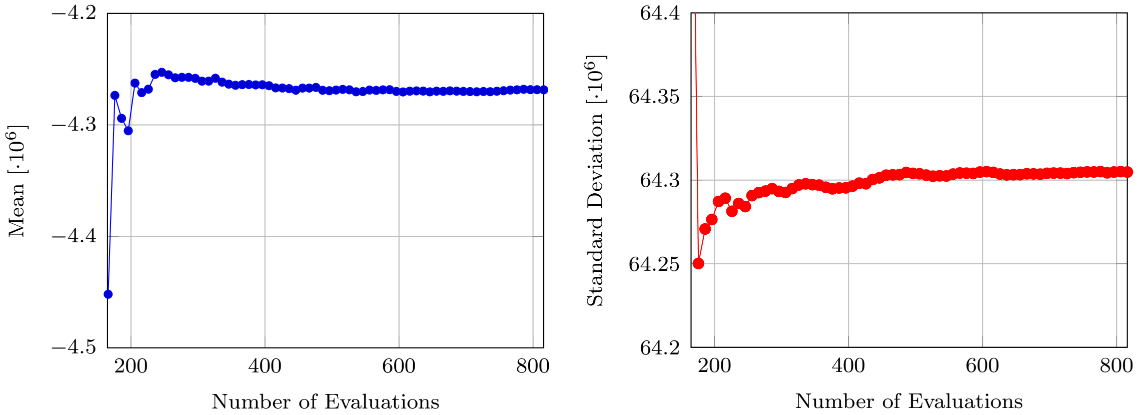

Figure 4.3: *Problem 3: NSPCE via OLS convergence for $k = 3$, using 165 polynomials: mean (left) and standard deviation (right) plotted against the increasing number of evaluations.*

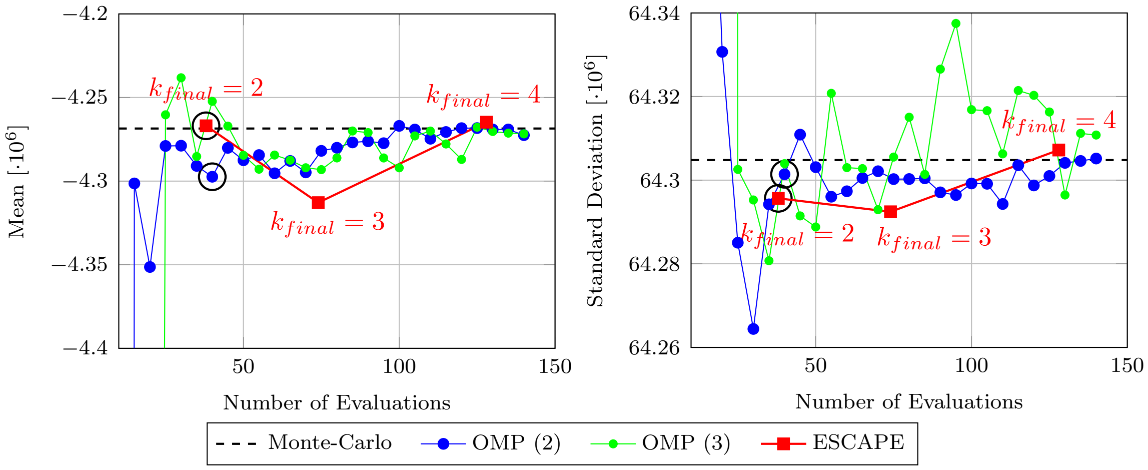

Figure 4.4: *Problem 3: Comparison of the convergence of the mean (left) and standard deviation (right) between OMP with $k = 2$ (blue), OMP with $k = 3$ (green), and ESCAPE, annotated with the corresponding initialization parameter k_{final}.*

Table 4.2 presents the final results for this case study which are also circled in Figures 4.4 and 4.5. Both OMP and ESCAPE seem to achieve very close estimates of the mean and standard deviation. Notably, ESCAPE achieves this while employing the smallest number of polynomial terms and nearly the same number of function evaluations as OMP, yet with slightly higher accuracy.

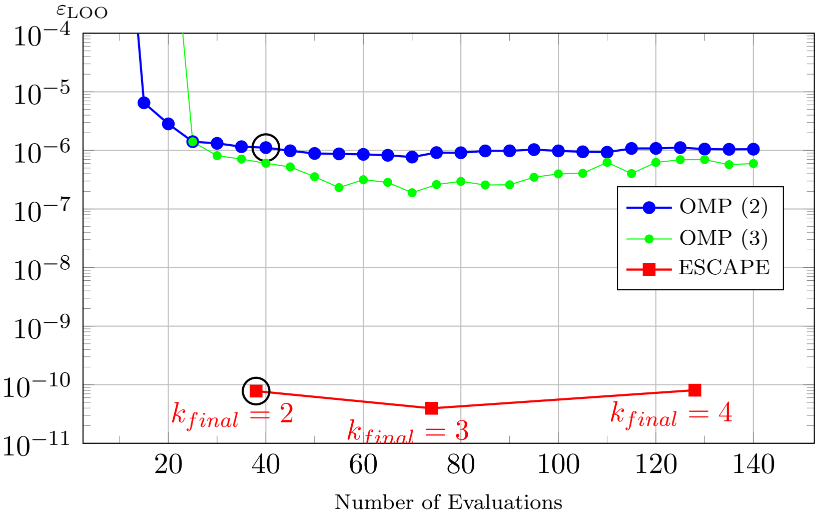

Figure 4.5: *Problem 3: ε_{LOO} error for the OMP and ESCAPE methods using different numbers of evaluations. ESCAPE results are annotated with the corresponding initialization parameter k_{final}.*

Method (k)	$\mu_{\hat{F}}$ $[\cdot 10^6]$	$\sigma_{\hat{F}}$ $[\cdot 10^6]$	P_{sparse}/CR	Cost (Function Calls)
NSPCE (3)	**-4.2685419237**	**64.3048095939**	-	**816**
OMP (2)	-4.2973995039	64.3014172751	15/0.3	40
ESCAPE (2)	-4.2669236655	64.2956551353	8/0.17	38

Table 4.2: *Problem 3: Comparison of the OMP and ESCAPE methods for estimating the mean μ and standard deviation σ of the QoI, compared against the reference NSPCE for $k = 3$. The table also presents the P_{sparse}, the CR, and the number of function evaluations (Cost).*

4.2 Problem 4: Bend Bifurcation Duct

Next, a bend bifurcation duct is selected to compare how the ESCAPE and OMP methods handle $D = 25$ uncertain variables related to geometry and flow conditions. The comparison is performed by computing the mean and standard deviation of the flow rate at one of the two outlets. For $k = 3$, the total number of non-sparse polynomials is $P_{\text{total}} = 351$.

Model Description

The velocity boundary conditions are defined with a uniform inflow of 1 m/s, while both outlet boundaries are assigned a zero-gradient condition to allow free outflow. No-slip conditions are imposed along all solid walls. The computational domain represents a 2D, bend bifurcation duct geometry. Two key geometric dimensions are the inlet width, approximately 0.1 m, and the total length of the duct in the streamwise direction, 0.3 m. The computational mesh, shown in Figure 4.6, consists of 15,000 hexahedral cells.

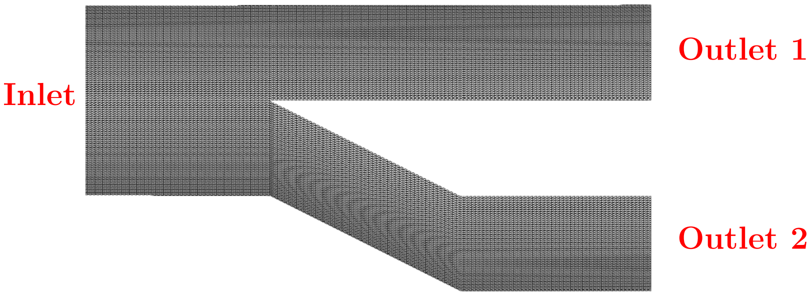

Figure 4.6: *Problem 4: Computational mesh.*

At both outlet branches, zero-gradient velocity and fixed static pressure conditions (set to identical values) are applied. No-slip boundary conditions are enforced on all walls. The flow is assumed to be laminar with a kinematic viscosity of $\nu = 10^{-3}\,\mathrm{m}^2/\mathrm{s}$.

The objective function is defined as the volume flow rate through the 'Outlet 1' patch shown in Figure 4.6. This is computed as the surface integral of the velocity vector projected along the outward normal of the outlet surface. The UQ aims to calculate the mean and standard deviation of this QoI, which is expressed mathematically as:

$$J = \int_{A_{\mathrm{Outlet1}}} \mathbf{v} \cdot \mathbf{n} \, dA,$$

where $\mathbf{v}$ is the velocity field and $\mathbf{n}$ is the unit outward normal vector on the outlet surface A_{Outlet1}.

As shown in Figure 4.7, the morphing box displays the CPs parametrized using B-splines, similar to the previous case. Red CPs (1–12), together with the inlet velocity, are subject to normally distributed uncertainties, as listed in Table 4.3, while blue CPs remain fixed to preserve continuity of the mesh and of second-order derivatives.

As illustrated in Figure 4.8, the convergence behavior of the reference Monte-Carlo method is clearly demonstrated.

Figure 4.9 presents the mean and standard deviation obtained using the two methods

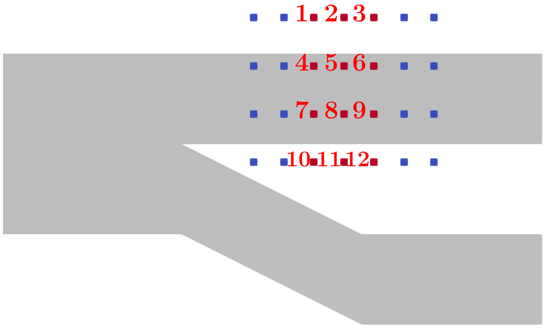

Figure 4.7: *Problem 4: Morphing box of the bend bifurcation duct.*

Uncertain Parameters:	Mean (μ)	Standard Dev. (σ)
Displacements Δx, Δy of CPs 1–12 (24 inputs)	0 m	0.01 m
Inlet velocity	1 m/s	0.2 m/s

Table 4.3: *Problem 4: Mean and standard deviation of the normal distributions used for the displacement of CPs 1–12 and inlet velocity.*

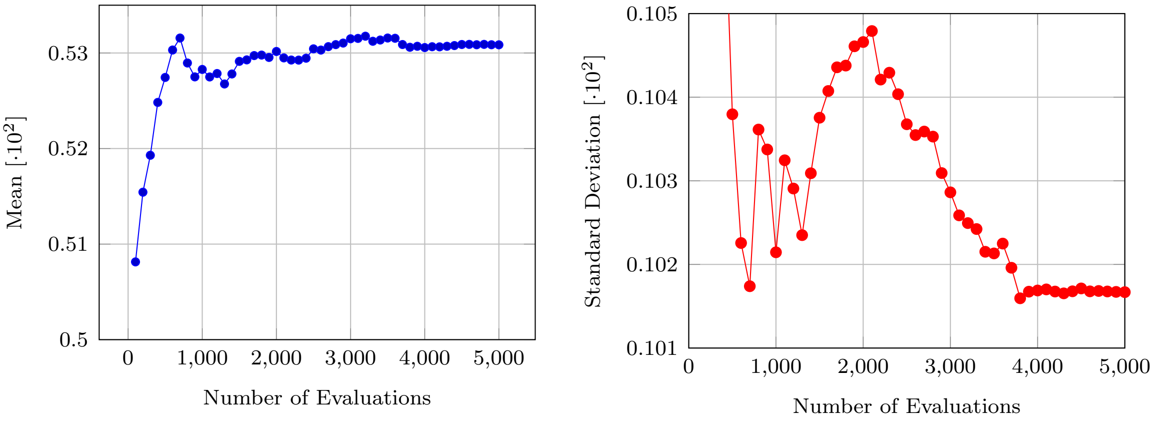

Figure 4.8: *Problem 4: Monte-Carlo convergence of the mean (left) and standard deviation (right) with increasing number of evaluations.*

under study, namely the ESCAPE and OMP. The OMP method is shown using polynomial orders up to $k = 2$ (in blue) and $k = 3$ (in green). The ESCAPE method, which falls almost immediately to the converged values, is presented for $k_{final} = 2$ and $k_{final} = 3$, demonstrating comparable accuracy and stability. Notably, the ESCAPE method starts producing reliable estimates at roughly the same point where the OMP method converges across various sample sizes, suggesting that the selected oversampling ratio is well matched to the problem's convergence behaviour.

Figure 4.10 shows $\varepsilon_{\mathrm{LOO}}$ for the OMP and ESCAPE methods, both of which attain low values, with their predictions are considered reliable.

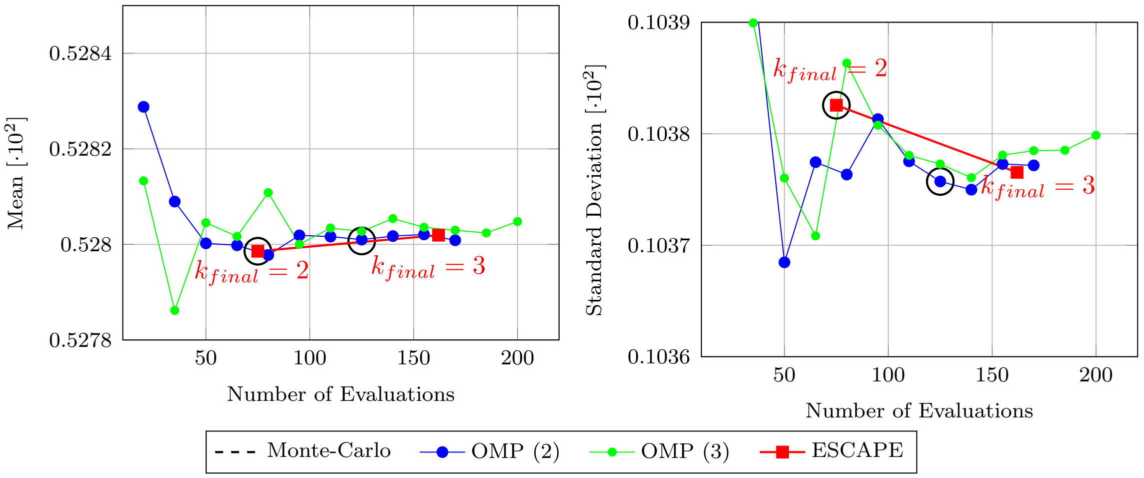

Figure 4.9: *Problem 4: Comparison of the convergence of the mean (left) and standard deviation (right) between OMP with $k = 2$ (blue), OMP with $k = 3$ (green), and ESCAPE.*

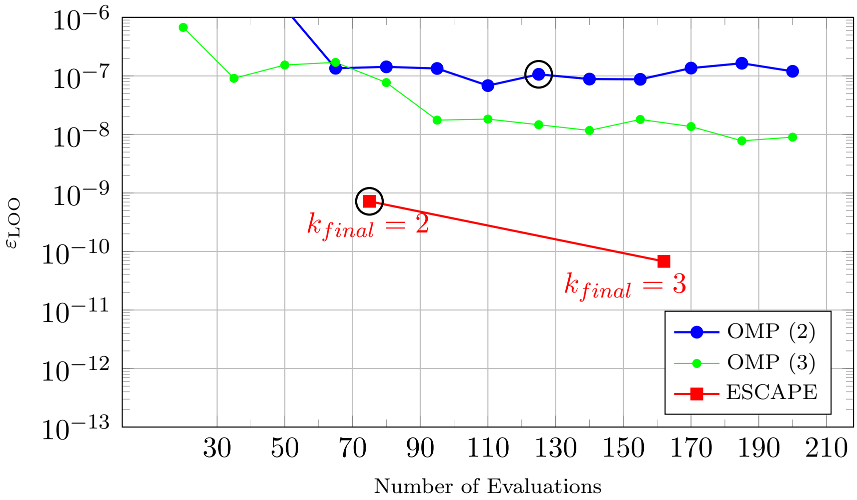

Figure 4.10: *Problem 4: ε_{LOO} error for the OMP and ESCAPE methods using different numbers of evaluations.*

Table 4.4 presents the final results for this case study. These selected cases, highlighted (circled) in Figures 4.9 and 4.10, are used to evaluate the relative performance of the two methods. Both methods produce results close to the Monte-Carlo values. Notably, ESCAPE achieves similar accuracy while selecting significantly less polynomial terms (11 vs. 62 for OMP) and requiring less model evaluations (75 vs. 125), demonstrating its efficiency in achieving a sparse representation with reduced computational cost.

Method (k)	$\mu_{\hat{F}}$ [$\cdot 10^2$]	$\sigma_{\hat{F}}$ [$\cdot 10^2$]	P_{sparse}/CR	Cost (Function Calls)
Monte-Carlo	**0.5308556934**	**0.1016684421**	**-**	**5000**
OMP (2)	0.5280100220	0.1037572010	62/0.18	125
ESCAPE (2)	0.5279857390	0.1038256064	11/0.03	75

Table 4.4: *Problem 4: Comparison of the OMP and ESCAPE methods in estimating the mean μ and standard deviation σ of the QoI. The table also presents the P_{sparse}, the compression ratio (CR), and the number of function evaluations, alongside the Monte-Carlo reference results.*

4.3 Problem 5: Bend Duct

As the final case study, an S-shaped bend duct is examined. Due to its large morphing region, this duct is particularly well-suited for evaluating UQ methods under a high number of geometric input variables. In this study, $D = 50$ geometric uncertainties are introduced to rigorously assess and compare the performance of the ESCAPE and OMP methods. For $k = 2$, the total number of non-sparse polynomials is $P_{\text{total}} = 1326$.

Model Description

This study examines the development of a laminar flow within a 2D, S-shaped duct Figure 4.11. The cross-sectional height of the duct is approximately 0.38 m, while its total length measures around 6.8 m. The computational domain is discretized using a structured mesh comprising 24,000 hexahedral cells, providing sufficient resolution to capture the flow characteristics within the bend geometry.

Regarding the flow conditions, a Reynolds number of Re = 1000 (laminar) is selected. The inlet velocity is prescribed as $\mathbf{v} = (0.039473, 0, 0)\,\text{m/s}$, i.e., aligned with the x-axis, while the static pressure at the outlet is fixed to $p = 0\,\text{Pa}$. The fluid has a kinematic viscosity of $\nu = 1.5 \cdot 10^{-5}\,\text{m}^2/\text{s}$.

The QoI is the volume-weighted total pressure loss:

$$J \;=\; -\int_{S_{I,O}} \left(p + \tfrac{1}{2}\rho\,\mathbf{v}\cdot\mathbf{v}\right)\,\mathbf{v}\cdot\mathbf{n}\,\mathrm{d}S, \tag{4.1}$$

where $\mathbf{v}$ is the velocity field and $\mathbf{n}$ the outward-pointing surface normal.

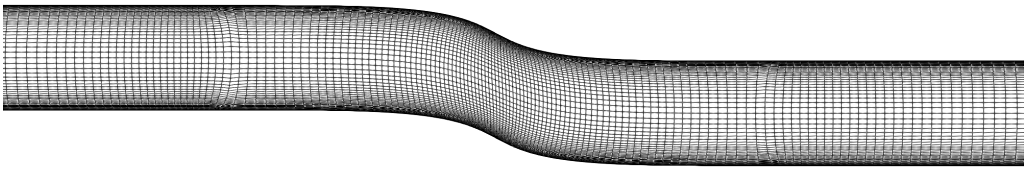

Figure 4.11: *Problem 5: Computational mesh.*

For shape parameterization, volumetric B-spline-based Free-Form Deformation is employed. In order to maintain continuity, the stationary part of the mesh must be preserved by keeping the boundary CPs and their neighboring points (blue points in Figure 4.12) constant.

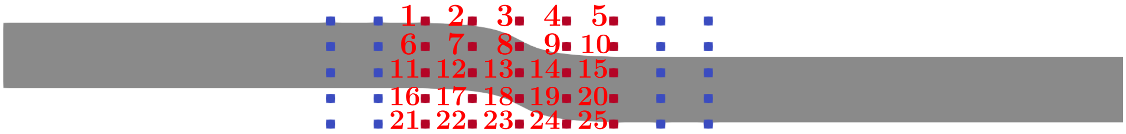

Figure 4.12: *Problem 5: Morphing box of the bend duct*

The uncertain inputs in this example correspond to the displacements of CPs 1-25, as shown in Table 4.5, all of which are normally distributed.

In Figure 4.13, the convergence characteristics of the reference Monte-Carlo method are clearly illustrated. Figure 4.14 illustrates the computed mean and standard deviation using the OMP method. In Figure 4.15, the ϵ_{LOO} of the OMP and the ESCAPE method is presented. ESCAPE exhibits a slightly higher ε_{LOO} than OMP.

Uncertain Parameters:	**Mean (μ)**	**Standard Dev. (σ)**
Displacements $\Delta x, \Delta y$ of CPs 1–25 (50 inputs)	0 m	0.1 m

Table 4.5: *Problem 5: Mean and standard deviation of the normal distributions used for the displacement of the x and y coordinate of CPs 1–25.*

However, based on Monte-Carlo and the observed convergence between $k_{final} = 2$ and $k_{final} = 3$ for ESCAPE, the results can still be considered reliable despite the marginally higher ε_{LOO}.

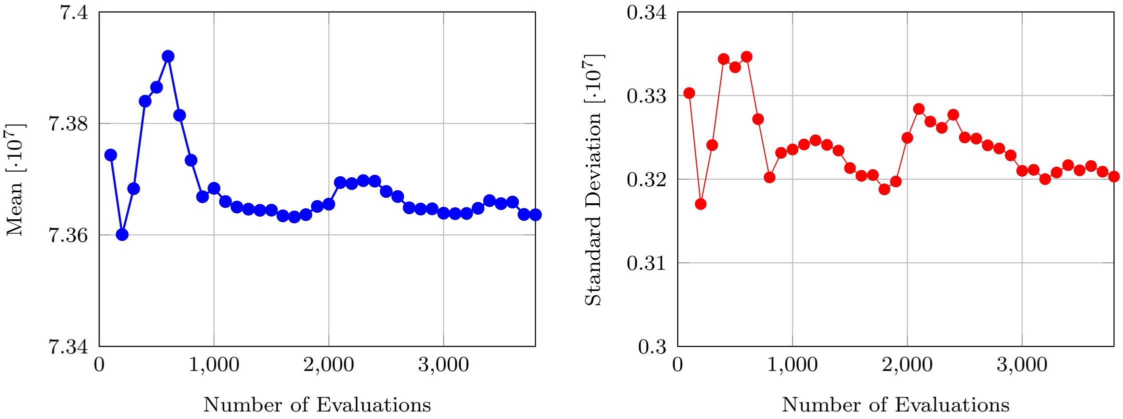

Figure 4.13: *Problem 5: Convergence of Monte-Carlo estimates for the mean (left) and standard deviation (right).*

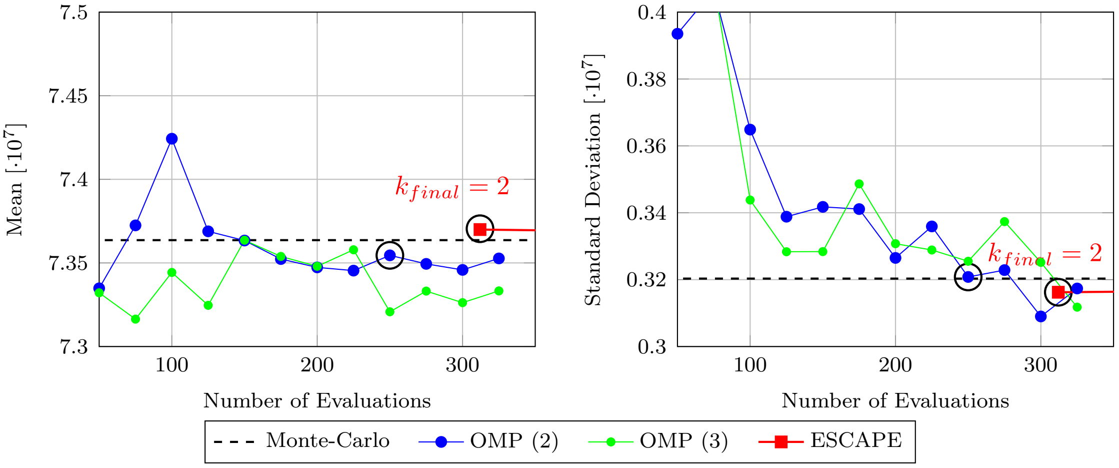

Figure 4.14: *Problem 5: Comparison of the convergence of the mean (left) and standard deviation (right) obtained with OMP and ESCAPE.*

The final results for this case study are summarized in Table 4.6. Both methods seem to estimate the mean and standard deviation close to the Monte-Carlo results. Although the LOO errors are relatively higher than in previous cases, both approaches still predict the mean and standard deviation reasonably well. ESCAPE

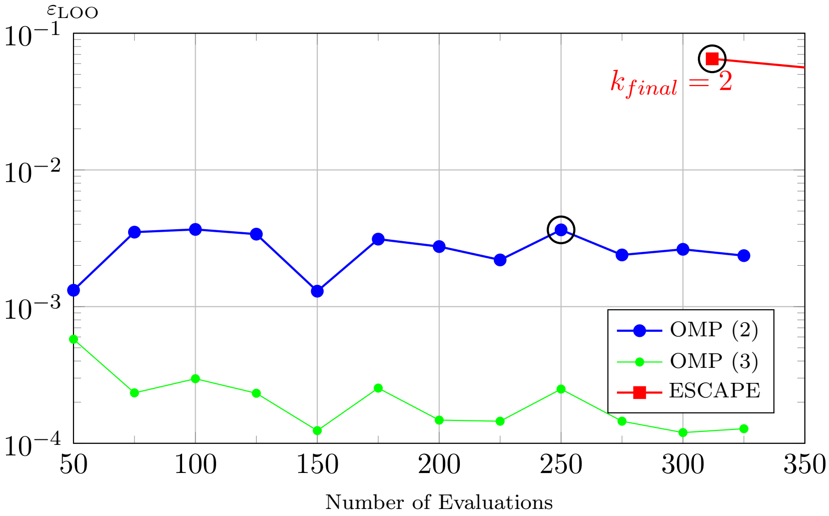

Figure 4.15: *Problem 5: Comparison of $\varepsilon_{\mathrm{LOO}}$ for the OMP and ESCAPE methods as a function of the number of function evaluations.*

achieves a sparser polynomial representation by selecting less terms (70 vs. 125 for OMP), while requiring 25% more function evaluations (312 vs. 250). The higher cost of ESCAPE, in this case, probably indicates the need for specific algorithmic tuning in high-dimensional settings such as the present one, where $D = 50$.

Method (k)	$\mu_{\hat{F}}$ [$\cdot 10^7$]	$\sigma_{\hat{F}}$ [$\cdot 10^7$]	P_{sparse}/CR	Cost (Function Calls)
Monte-Carlo	**7.3636198298**	**0.3203129694**	**-**	**3800**
OMP (2)	7.3545298325	0.3208058326	125/0.09	250
ESCAPE (2)	7.3699597468	0.3162349518	70/0.05	312

Table 4.6: *Problem 5: Comparison of the OMP and ESCAPE methods for estimating the mean μ and standard deviation σ of the QoI. The table also presents the P_{sparse}, CR, and the number of function evaluations, alongside the Monte-Carlo reference results.*

Chapter 5

Conclusions and Recommendations for Future Work

5.1 Concluding Remarks

This thesis addresses multi-dimensional UQ problems, tested with up to 50 uncertain inputs, by programming and evaluating two cost-effective, sparse regression-based PCE methods: the well-known OMP method and the ESCAPE method, which is proposed in the context of this MSc thesis and inspired by existing approaches.

Figures 5.1 and 5.2 summarize the performance of OMP and ESCAPE across all five problems, in terms of relative errors in the mean and standard deviation, as well as computational cost. The relative errors are computed with respect to a high-fidelity reference solution, expressed as a percentage, allowing a direct comparison of how accurately each method reproduces the mean and standard deviation of the QoIs. In Figure 5.1, the left panel illustrates that both methods achieve relative errors in the mean of less than 1%, with ESCAPE slightly outperforming OMP in Problem 3.The right panel presents relative errors in the standard deviation, showing comparable performance between the two methods for most problems; however, ESCAPE exhibits a larger error for Problem 5 ($D = 50$ stochastic inputs). Overall, both methods remain highly accurate, with relative errors below 2.2%. Figure 5.2 presents the computational cost in terms of function calls. In some cases, ESCAPE shows an advantage over the OMP algorithm, whose linear cost growth can lead to an overestimation of the required samples (e.g., when $N = 5D$ is excessive). For instance, ESCAPE can achieve similar or lower computational costs for small- to medium-scale problems (Problems 1–4, $D = 8$–25). In other cases, such as Problem 5 with $D = 50$, the computational cost of ESCAPE is higher, where OMP demonstrates better accuracy and cost-reduction. This suggests that further tuning of ESCAPE may be necessary to handle problems of this dimensionality more efficiently. While OMP and ESCAPE maintain manageable computational costs with an approximately linear trend as the number of stochastic inputs increases, NSPCE ($k = 2$, SR $= 3$) becomes practically intractable in high-dimensional problems due to its exponentially growing cost. This underscores the crucial role of OMP and ESCAPE in enabling scalable, high-dimensional UQ.

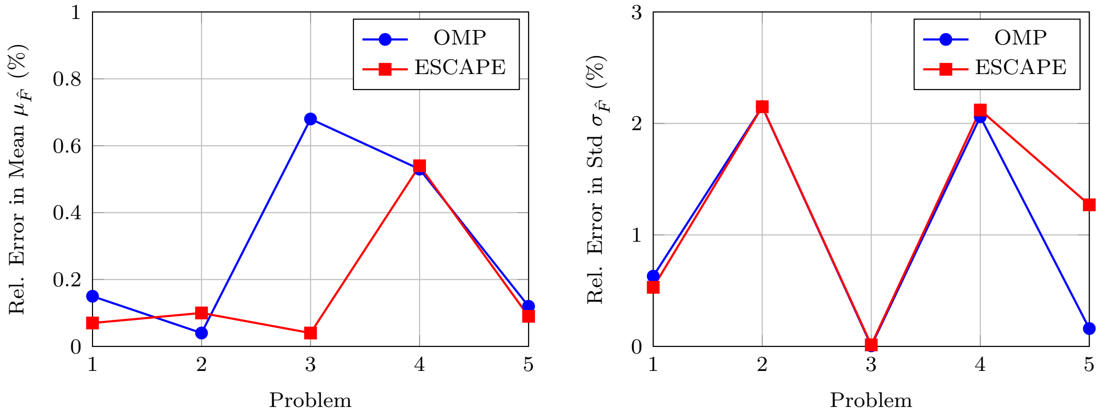

Figure 5.1: *Comparison of OMP vs ESCAPE relative errors across problems. Left: relative errors in mean, Right: relative errors in standard deviation.*

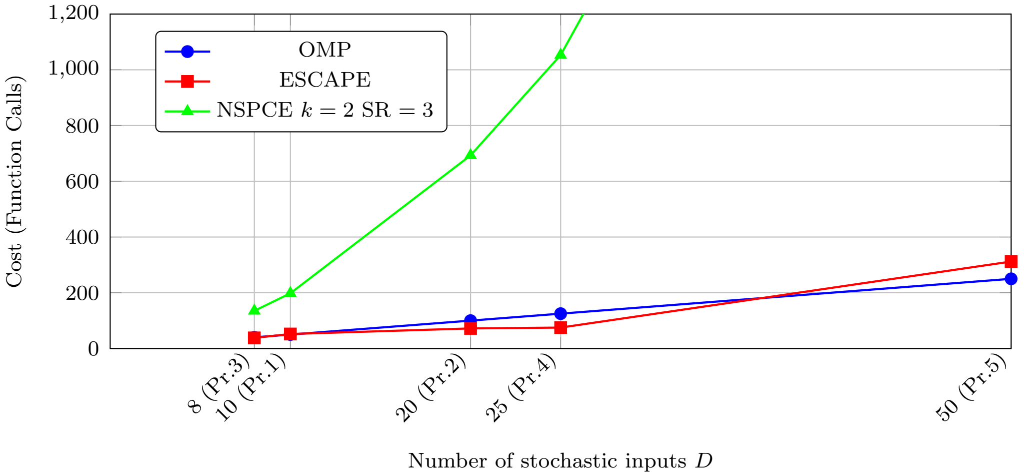

Figure 5.2: *Comparison of computational cost (measured in function calls) for OMP, ESCAPE, and NSPCE ($k = 2$, $SR = 3$) across problems with increasing dimension D. Problem numbers corresponding to each dimension are indicated in parentheses on the x-axis.*

5.2 Recommendations for Future Work

Building upon the findings of this study, future research is encouraged to explore various directions in the field of UQ. A potential direction for future work is the extension of sparse, regression-based PCE to handle input uncertain variables with arbitrary distributions. In such cases, standard polynomial families may not be applicable. One approach is to apply an isoprobabilistic transform, mapping the inputs to a space with independent standard marginals, which allows the use of classical polynomials but may introduce additional non-linearity and affect accuracy. Alternatively, custom orthonormal polynomials can be constructed directly. This strategy is more flexible and avoids transformation-induced distortions, although it requires additional computational effort. Implementing these approaches would broaden the applicability of sparse regression PCE [16].

Additionally, alternative algorithms for sparse PCE, such as Least Angle Regression (LARS) [8], could be implemented and systematically compared with existing methods like OMP and ESCAPE. Furthermore, incorporating algorithms that support both forward selection and backward elimination—i.e., those capable of not only adding significant polynomial terms to the basis but also removing polynomial terms added in previous iterations that lose their relevance—could enhance model sparsity and accuracy in an adaptive and efficient manner.

In this thesis, samples were generated using a C++ implementation of a standard normal random number generator. Future work could focus on assessing sample quality and selecting new points based on model uncertainty or error indicators, particularly in regions of poor surrogate accuracy. Such adaptive sampling strategies could yield more accurate PCE models with lower undersampling ratios and fewer costly function evaluations, thereby enhancing computational efficiency.

Appendix A'

Orthogonal Polynomials

A.1 Hermite Polynomials and Their Properties

Hermite polynomials are a family of orthogonal polynomials widely used in PCE for stochastic variables with normal distributions. There are two types of Hermite polynomials: the probabilists' Hermite polynomials, which are commonly used in statistics, and the physicists' Hermite polynomials, which are preferred in physics. This section focuses exclusively on the probabilists' Hermite polynomials.

The probabilists' Hermite polynomials are defined as uniparametric functions in the domain $(-\infty, \infty)$:

$$H_k(x) = (-1)^k e^{\frac{x^2}{2}} \frac{d^k}{dx^k} e^{-\frac{x^2}{2}}. \tag{A.1}$$

These polynomials satisfy the following recurrence relation:

$$H_{k+1}(x) = xH_k(x) - kH_{k-1}(x). \tag{A.2}$$

Probabilistic Hermite Polynomials

The first ten polynomials in this sequence are given by:

$$H_0(x) = 1, \tag{A.3}$$
$$H_1(x) = x, \tag{A.4}$$
$$H_2(x) = x^2 - 1, \tag{A.5}$$
$$H_3(x) = x^3 - 3x, \tag{A.6}$$
$$H_4(x) = x^4 - 6x^2 + 3, \tag{A.7}$$
$$H_5(x) = x^5 - 10x^3 + 15x, \tag{A.8}$$
$$H_6(x) = x^6 - 15x^4 + 45x^2 - 15, \tag{A.9}$$
$$H_7(x) = x^7 - 21x^5 + 105x^3 - 105x, \tag{A.10}$$
$$H_8(x) = x^8 - 28x^6 + 210x^4 - 420x^2 + 105, \tag{A.11}$$
$$H_9(x) = x^9 - 36x^7 + 378x^5 - 1260x^3 + 945x. \tag{A.12}$$

The derivative of a Hermite polynomial follows:

$$\frac{d}{dx} H_k(x) = k H_{k-1}(x). \tag{A.13}$$

The probabilists' Hermite polynomials are orthogonal with respect to the weight function on the domain $(-\infty, \infty)$:

$$w(x) = e^{-x^2/2}, \tag{A.14}$$

which gives the orthogonality condition:

$$\int_{-\infty}^{\infty} H_m(x) H_k(x) e^{-x^2/2} dx = k! \sqrt{2\pi} \delta_{mk}. \tag{A.15}$$

These properties make Hermite polynomials fundamental tools in UQ and PCE for modeling normally distributed random variables.

Appendix B'

LOO cross validation

B.1 Proof of the LOO Error Formula

Notation and Problem Statement

Consider a PCE model of order k:

$$\hat{F}(\mathbf{x}) = \sum_{j=0}^{P-1} \alpha_j \psi_j(\mathbf{x}). \tag{B.1}$$

Let $\boldsymbol{f} \in \mathbb{R}^N$ be the vector of model evaluations at the sample points:

$$\boldsymbol{f} = \begin{bmatrix} F(\mathbf{x}_0) \\ \vdots \\ F(\mathbf{x}_{N-1}) \end{bmatrix}$$

Predicted Error

The predicted error is defined as the difference between the model evaluation at $\mathbf{x}_i$ and its prediction based on $\hat{F}_{\mathbf{X}\setminus i}(\mathbf{x}_i)$:

$$\Delta^{(i)} = F(\mathbf{x}_i) - \hat{F}_{\mathbf{X}\setminus i}(\mathbf{x}_i). \tag{B.2}$$

Design Matrices

Define the full design matrix $\boldsymbol{\Psi} \in \mathbb{R}^{N \times P}$:

$$\boldsymbol{\Psi} = \begin{pmatrix} \psi_0(\mathbf{x}_0) & \cdots & \psi_{P-1}(\mathbf{x}_0) \\ \vdots & \ddots & \vdots \\ \psi_0(\mathbf{x}_{N-1}) & \cdots & \psi_{P-1}(\mathbf{x}_{N-1}) \end{pmatrix}.$$

The reduced design matrix $\boldsymbol{\Psi}_i \in \mathbb{R}^{(N-1) \times P}$ excludes the i-th row:

$$\boldsymbol{\Psi}_i = \begin{pmatrix} \psi_0(\mathbf{x}_0) & \cdots & \psi_{P-1}(\mathbf{x}_0) \\ \vdots & \ddots & \vdots \\ \psi_0(\mathbf{x}_{i-1}) & \cdots & \psi_{P-1}(\mathbf{x}_{i-1}) \\ \psi_0(\mathbf{x}_{i+1}) & \cdots & \psi_{P-1}(\mathbf{x}_{i+1}) \\ \vdots & \ddots & \vdots \\ \psi_0(\mathbf{x}_{N-1}) & \cdots & \psi_{P-1}(\mathbf{x}_{N-1}) \end{pmatrix}.$$

Coefficient Formulations

The PCE coefficients for the full model are obtained by solving the system via OLS:

$$\boldsymbol{\alpha} = (\boldsymbol{\Psi}^T \boldsymbol{\Psi})^{-1} \boldsymbol{\Psi}^T \boldsymbol{f} = \mathbf{M}^{-1} \boldsymbol{\Psi}^T \boldsymbol{f}, \tag{B.3}$$

where $\mathbf{M} = \boldsymbol{\Psi}^T \boldsymbol{\Psi}$.

For the reduced model (excluding the i-th point), the coefficients are:

$$\boldsymbol{\alpha}_i = \mathbf{M}_i^{-1} \boldsymbol{\Psi}_i^T \boldsymbol{f}_i, \tag{B.4}$$

where $\mathbf{M}_i = \boldsymbol{\Psi}_i^T \boldsymbol{\Psi}_i$ and $\boldsymbol{f}_i$ is $\boldsymbol{f}$ without the i-th element.

Relationship Between $\mathbf{M}_i^{-1}$ and $\mathbf{M}^{-1}$ Matrices

It can be shown that $\mathbf{M}_i^{-1}$ is related to its counterpart $\mathbf{M}^{-1}$ as follows:

$$\mathbf{M}_i^{-1} = (\mathbf{M} - \boldsymbol{\psi}_i \boldsymbol{\psi}_i^T)^{-1}, \tag{B.5}$$

using the matrix inversion lemma (Sherman–Morrison–Woodbury identity) [13]:

$$(\mathbf{M} - \boldsymbol{\psi}_i \boldsymbol{\psi}_i^T)^{-1} = \mathbf{M}^{-1} + \frac{\mathbf{M}^{-1} \boldsymbol{\psi}_i \boldsymbol{\psi}_i^T \mathbf{M}^{-1}}{1 - h_i}, \tag{B.6}$$

where

$$\boldsymbol{\psi}_i = (\psi_0(\mathbf{x}_i),\ \ldots,\ \psi_{P-1}(\mathbf{x}_i))^T\,,$$

is the i-th row of $\boldsymbol{\Psi}_i$, and

$$h_i = \boldsymbol{\psi}_i^T \mathbf{M}^{-1} \boldsymbol{\psi}_i.$$

LOO Prediction

The LOO prediction can be expressed as:

$$\begin{aligned}
\hat{F}_{\mathbf{X}\setminus i}(\mathbf{x}_i) &= \boldsymbol{\psi}_i^T \boldsymbol{\alpha}_i && \text{(B.7)}\\
&= \boldsymbol{\psi}_i^T \left(\mathbf{M}^{-1} + \frac{\mathbf{M}^{-1}\boldsymbol{\psi}_i \boldsymbol{\psi}_i^T \mathbf{M}^{-1}}{1-h_i} \right) \left(\boldsymbol{\Psi}^T \boldsymbol{f} - F(\mathbf{x}_i)\boldsymbol{\psi}_i \right) && \text{(B.8)}\\
&= \hat{F}(\mathbf{x}_i) - \frac{h_i}{1-h_i}(F(\mathbf{x}_i) - \hat{F}(\mathbf{x}_i)). && \text{(B.9)}
\end{aligned}$$

.

Final LOO Formula

Substituting into the LOO error definition:

$$\begin{aligned}
\Delta^{(i)} &= F(\mathbf{x}_i) - \hat{F}_{\mathbf{X}\setminus i}(\mathbf{x}_i) && \text{(B.10)}\\
&= F(\mathbf{x}_i) - \left[\hat{F}(\mathbf{x}_i) - \frac{h_i}{1-h_i}(F(\mathbf{x}_i) - \hat{F}(\mathbf{x}_i)) \right] && \text{(B.11)}\\
&= \frac{F(\mathbf{x}_i) - \hat{F}(\mathbf{x}_i)}{1-h_i}. && \text{(B.12)}
\end{aligned}$$

LOO Error Estimate

The LOO error according to [5] is therefore:

$$\varepsilon_{\mathrm{LOO}} = \frac{1}{N} \sum_{i=0}^{N-1} \left(\frac{F(\mathbf{x}_i) - \hat{F}(\mathbf{x}_i)}{1-h_i} \right)^2. \tag{B.13}$$

This formula allows computation of the LOO error without constructing N separate models, using only the residuals and leverage scores from the full model.

Bibliography

[1] Ahlfeld, R., Belkouchi, B., Montomoli, F.: SAMBA: Sparse Approximation of Moment-Based Arbitrary Polynomial Chaos. Journal of Computational Physics (2016)

[2] Askey, R., Wilson, J.: Some Basic Hypergeometric Polynomials that Generalize Jacobi Polynomials, Memoirs of the American Mathematical Society, vol. 54. Providence, RI (1985)

[3] Berveiller, M., Sudret, B., Lemaire, M.: Stochastic finite elements: a non-intrusive approach by regression. European Journal of Mechanics - A/Solids (2006)

[4] Blatman, G., Sudret, B.: Adaptive sparse polynomial chaos expansion based on least angle regression. Journal of Computational Physics (2011)

[5] Blatman, G.: Adaptive Sparse Polynomial Chaos Expansions for Uncertainty Propagation and Sensitivity Analysis. Ph.D. thesis, Université Blaise Pascal - Clermont II, Clermont-Ferrand, France (2009)

[6] Cohen, A., Migliorati, G.: Multivariate approximation in downward closed polynomial spaces (2016)

[7] Dodson, M., Parks, G.: Robust aerodynamic design optimization using polynomial-chaos. Journal of Aircraft (2009)

[8] Efron, B., Hastie, T., Johnstone, I., Tibshirani, R.: Least angle regression. The Annals of Statistics (2004)

[9] Eldred, M.S., Burkardt, J.: Comparison of non-intrusive polynomial chaos and stochastic collocation methods for uncertainty quantification. Proceedings of the 47th AIAA Aerospace Sciences Meeting (2009)

[10] Forrester, A.I.J., Sobester, A., Keane, A.J.: Engineering Design via Surrogate Modelling: A Practical Guide. Chichester, UK (2008)

[11] Geisser, S.: The predictive sample reuse method with applications. Journal of the American Statistical Association (1975)

[12] Giannakoglou, K.C., Anagnostopoulos, I., Bergeles, G.: Numerical Analysis for Engineers. National Technical University of Athens, 3rd edn. (2003)

[13] Golub, G.H., Van Loan, C.F.: Matrix Computations. Johns Hopkins University Press, 4th edn. (2013)

[14] Loukrezis, D., Diehl, E., Gersem, H.D.: Multivariate sensitivity-adaptive polynomial chaos expansion for high-dimensional surrogate modeling and uncertainty quantification. Applied Mathematical Modelling (2025)

[15] Mallat, S.G., Zhang, Z.: Matching pursuits with time-frequency dictionaries. IEEE Transactions on Signal Processing (1993)

[16] Marelli, S., Lüthen, N., Sudret, B.: UQLab User Manual – Polynomial Chaos Expansions (2024)

[17] Molinaro, A.M., Simon, R., Pfeiffer, R.M.: Prediction error estimation: a comparison of resampling methods. Bioinformatics (2005)

[18] Najm, H.N.: Uncertainty quantification and polynomial chaos techniques in computational fluid dynamics. Annual Review of Fluid Mechanics (2009)

[19] Pampalis, G.: Implementation of Polynomial Chaos Expansion in Aerodynamic Robust Design - Optimization with Evolutionary Algorithms under Stochastic Inputs. Master's thesis, National Technical University of Athens (2015)

[20] Pati, Y.C., Rezaiifar, R., Krishnaprasad, P.S.: Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. Proceedings of the 27th Asilomar Conference on Signals, Systems and Computers (1993)

[21] Poles, S., Lovinson, A.: A polynomial chaos approach to robust multiobjective optimization. In: Hybrid and Robust Approaches to Multiobjective Optimization (2009)

[22] Raymer, D.P.: Aircraft Design: A Conceptual Approach. Reston, VA (1992)

[23] Sobol, I.: Sensitivity estimates for nonlinear mathematical models. Mathematical Modelling and Computational Experiments (1993)

[24] Sobol, I.: Global sensitivity indices for nonlinear mathematical models and their monte carlo estimates. Mathematics and Computers in Simulation (2001)

[25] Stone, M.: Cross-validatory choice and assessment of statistical predictions. Journal of the Royal Statistical Society: Series B (Methodological) (1974)

[26] Sudret, B.: Uncertainty propagation and sensitivity analysis in mechanical models - Contributions to structural reliability and stochastic spectral methods. Ph.D. thesis, Université Blaise Pascal, Clermont-Ferrand, France (2007)

[27] Wiener, N.: The homogeneous chaos. American Journal of Mathematics (1938)

[28] Xiu, D.: Fast numerical methods for stochastic computations. Communications in Computational Physics (2009)

[29] Xiu, D., Karniadakis, G.E.: The Wiener–Askey polynomial chaos for stochastic differential equations. SIAM Journal on Scientific Computing (2002)