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

Sparse Polynomial Chaos Expansions for Uncertainty Quantification in Aerodynamic Problems

Diploma Thesis

Titos - Georgios Dellis

Advisor: Kyriakos C. Giannakoglou, Professor NTUA

Athens, September 2025

Acknowledgments

First and foremost, I would like to express my deepest gratitude to Professor Kyriakos Giannakoglou for his invaluable guidance, inspiration, and continuous support throughout the course of this thesis. His insights, encouragement, and trust were instrumental not only in shaping the technical direction of this work but also in developing my understanding of research in computational science and engineering.

I am also sincerely thankful to Dr. Varvara Asouti for her generous assistance and expertise in the setup of the aerodynamic software used in this thesis. Her support in understanding the simulation environment and her practical advice greatly facilitated the successful execution of the aerodynamic applications.

Their combined mentorship played a crucial role in the completion of this thesis, and I am truly grateful for their time, patience, and willingness to help.

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

Sparse Polynomial Chaos Expansions for Uncertainty Quantification in Aerodynamic Problems

Diploma Thesis

Titos - Georgios Dellis

Advisor: Kyriakos C. Giannakoglou, Professor NTUA Athens, September 2025

Abstract

This thesis investigates the use of non-intrusive, regression-based Polynomial Chaos Expansions (PCE) for Uncertainty Quantification (UQ) in aerodynamic simulations. While regression PCE is more efficient than Monte Carlo methods, it suffers from the curse of dimensionality, as the number of polynomial terms (unknown PCE coefficients) grows rapidly with input dimension, leading to large demands in costly model evaluations. To address this limitation, the thesis employs sparse regression PCE techniques that exploit sparsity in the expansion coefficients to construct accurate surrogates with fewer samples.

Unlike Ordinary Least Squares (OLS), which requires oversampling (typically at least twice as many samples as unknown coefficients) to ensure accurate approximation, sparse regression can succeed in undersampled regimes where the number of samples is smaller than the polynomial basis size. Reducing the sample requirements is crucial, since sample evaluations via complex simulation models constitute the largest portion of computational cost in a PCE application.

Two widely used approaches for sparse PCE are considered: convex optimization methods such as LASSO and greedy algorithms such as Orthogonal Matching Pursuit (OMP). In addition, enhanced variants—post-LASSO and relaxed LASSO—are employed, which reduce the basis by first identifying the most relevant terms and then re-estimating their coefficients. Software was developed to implement these solvers (LASSO, OMP, post-LASSO, relaxed LASSO) in the context of regression

PCE, which were validated on a variant of the Borehole benchmark and two aerodynamic applications: (i) an XFOIL-based simulation of a NACA 2412 airfoil under uncertain operating conditions, and (ii) a CFD-based simulation of a NACA 16103 airfoil with geometric uncertainties represented via a Karhunen–Loève Expansion (KLE).

Results show that OLS requires oversampling by a factor of about two to achieve stable approximation. In contrast, LASSO and OMP provide reliable estimates with only 75–80% as many samples as unknown coefficients, while post-LASSO and relaxed LASSO maintain robust performance even when this ratio drops to around 65–70%. These findings demonstrate that incorporating sparsity and basis reduction into regression-based PCE substantially reduces computational cost while maintaining accuracy, establishing sparse PCE as an efficient framework for UQ in aerodynamics.

Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Μηχανολόγων Μηχανικών
Τομέας Ρευστών
Μονάδα Παράλληλης Υπολογιστικής Ρευστοδυναμικής & Βελτιστοποίησης

Αραιές Επεκτάσεις Πολυωνυμικού Χάους για την Ποσοτικοποίηση Αβεβαιότητας σε Αεροδυναμικά Προβλήματα

Διπλωματική Εργασία

Τίτος - Γεώργιος Δελλής

Επιβλέπων: Κυριάκος Γιαννάκογλου, Καθηγητής ΕΜΠ

Αθήνα, Σεπτέμβριος 2025

Περίληψη

Η παρούσα διπλωματική εργασία εξετάζει τη χρήση μη-επεμβατικών, βασισμένων σε παλινδρόμηση Επεκτάσεων Πολυωνυμικού Χάους (Polynomial Chaos Expansions , PCE) για την Ποσοτικοποίηση της Αβεβαιότητας (Uncertainty Quantification , UQ) σε αεροδυναμικές προσομοιώσεις. Αν και η μέθοδος PCE είναι αποδοτικότερη από τις μεθόδους Monte Carlo, υφίσταται τον «κατάρα της διαστατικότητας», καθώς ο αριθμός των πολυωνυμικών όρων (άγνωστοι συντελεστές PCE) αυξάνεται ραγδαία με τη διάσταση των εισόδων, οδηγώντας σε μεγάλο αριθμό απαιτούμενων αξιολογήσεων από υπολογιστικά δαπανηρά μοντέλα. Για την αντιμετώπιση αυτού του περιορισμού, η εργασία επικεντρώνεται σε τεχνικές αραιής παλινδρόμησης με PCE, οι οποίες εκμεταλλεύονται την αραιότητα των συντελεστών της επέκτασης ώστε να κατασκευάσουν ακριβή υποκατάστατα με μικρότερο αριθμό δειγμάτων.

Σε αντίθεση με την Κλασική Μέθοδο Ελαχίστων Τετραγώνων (Ordinary Least Squares , OLS), η οποία απαιτεί υπερδειγματοληψία (τυπικά περίπου διπλάσιο αριθμό δειγμάτων από άγνωστους συντελεστές) για να επιτευχθεί ακριβής λύση, οι μέθοδοι αραιής παλινδρόμησης μπορούν να επιτύχουν σε σενάρια όπου ο αριθμός δειγμάτων είναι μικρότερος από το μέγεθος της πολυωνυμικής βάσης. Η μείωση των απαιτήσεων σε δείγματα είναι

καθοριστικής σημασίας, καθώς οι αξιολογήσεις δειγμάτων μέσω πολύπλοκων προσομοιωτικών μοντέλων αποτελούν το μεγαλύτερο μέρος του υπολογιστικού κόστους σε μια εφαρμογή PCE.

Στην εργασία εξετάζονται δύο ευρέως χρησιμοποιούμενες προσεγγίσεις για αραιή PCE: η μέθοδος LASSO, που ανήκει στις κυρτές βελτιστοποιήσεις, και ο αλγόριθμος Orthogonal Matching Pursuit (OMP), που αποτελεί χαρακτηριστικό παράδειγμα greedyαλγορίθμου. Επιπλέον, μελετώνται βελτιωμένες εκδοχές όπως η μέθοδος post-LASSO και η παραλλαγή relaxed LASSO, οι οποίες μειώνουν τη βάση εντοπίζοντας αρχικά τους σημαντικότερους όρους και επανεκτιμώντας στη συνέχεια τους συντελεστές. Αναπτύχθηκε λογισμικό για την υλοποίηση των παραπάνω μεθόδων (μέθοδος LASSO, αλγόριθμος OMP, μέθοδος post-LASSO, παραλλαγή relaxed LASSO) στο πλαίσιο της παλινδρόμησης με PCE. Το λογισμικό επικυρώθηκε σε εφαρμογή της συνάρτησης Borehole και σε δύο αεροδυναμικές περιπτώσεις μελέτης: (ι) προσομοίωση με XFOIL της πτέρυγας NACA 2412 υπό αβέβαιες συνθήκες λειτουργίας και (ιι) προσομοίωση με CFD της πτέρυγας NACA 16103 με γεωμετρικές αβεβαιότητες, οι οποίες μοντελοποιούνται μέσω Karhunen–Loève Expansion (KLE).

Τα αποτελέσματα δείχνουν ότι η μέθοδος OLS απαιτεί περίπου διπλάσιο αριθμό δειγμάτων από άγνωστους συντελεστές για να επιτύχει σταθερή προσεγγιστική λύση. Αντίθετα, η μέθοδος LASSO και ο αλγόριθμος OMP παρέχουν αξιόπιστες εκτιμήσεις με μόλις 75–80% του αριθμού δειγμάτων σε σχέση με τους αγνώστους, ενώ οι εκδοχές post-LASSO και relaxed LASSO διατηρούν ικανοποιητική απόδοση ακόμη και όταν ο λόγος αυτός πέσει στο 65–70%. Τα ευρήματα αυτά καταδεικνύουν ότι η ενσωμάτωση αραιότητας και μείωσης βάσης στην παλινδρόμηση με PCE μειώνει σημαντικά το υπολογιστικό κόστος διατηρώντας παράλληλα την ακρίβεια, καθιστώντας τις μεθόδους αραιής PCE ένα αποδοτικό πλαίσιο για την Ποσοτικοποίηση της Αβεβαιότητας στην αεροδυναμική.

Contents

Nomenclature

BPDN	Basis Pursuit De-Noising
$\mathbf{C_D}$	Drag Coefficient
$\mathbf{C_L}$	Lift Coefficient
CDF	Cumulative Density Function
CFD	Computational Fluid Dynamics
CS	Compressed Sensing
DoE	Design of Experiments
ISTA	Iterative-Shrinkage-Thresholding-Algorithm
KLE	Karhunen-Loève Expansion
LASSO	Least Absolute Shrinkage and Selection Operator
LHS	Latin Hypercube Sampling
MC	Monte Carlo
$\mathbf{N(0,1)}$	Standard normal (Gaussian) distribution
OLS	Ordinary Least Squares
OMP	Orthogonal Matching Pursuit
PCE	Polynomial Chaos Expansion
PDF	Probability Density Function
QoI	Quantity of Interest
RIP	Restricted Isometry Property
RNG	Random Number Generator
SD	Standard Deviation
SR	Sampling Ratio
SVD	Singular Value Decomposition
UQ	Uncertainty Quantification

Symbols Table

ξ	Uncertain Input Variable
Ψ	Polynomial Basis
p	Chaos Order
N	Sample size
P	Basis size
u	Quantity of Interest
c	PCE Coefficient
A	Regression Matrix
λ	LASSO regularization parameter
k	Number of Non-Zero terms

Chapter 1

Introduction

1.1 Uncertainty Quantification of Deterministic Models

In engineering and applied sciences, numerical models are commonly used to describe physical systems and predict their behavior under user-defined conditions. For example, the flow field around an airfoil at known environmental conditions can be simulated using Computational Fluid Dynamics (CFD). The input of such a model includes the airfoil's geometry, the freestream velocity, the angle of attack, and the fluid's properties, such as viscosity, density, etc. After the simulation has been carried out, quantities of interest (QoI) can be computed from the flow field. These are usually integral quantities, such as the Lift-to-Drag ratio of the airfoil, which quantifies its aerodynamic performance under these fixed conditions.

While these models operate on deterministic input, real-world applications are subject to uncertainties due to measurement errors, material properties, environmental influences, and modeling assumptions. These uncertainties can propagate through the system and significantly affect the output, leading to discrepancies between predictions and actual performance. Therefore, if the input of the model actually shows stochastic behavior, the output (QoI) should also be modeled as an uncertain quantity.

Uncertainty Quantification (UQ) aims to assess the effect of these input uncertainties on the QoI. UQ typically involves three key steps: (1) identifying sources of uncertainty, (2) propagating uncertainties through the model, and (3) analyzing the statistical behavior of the output.

1.2 Polynomial Chaos Expansions for UQ

Traditional methods such as Monte Carlo (MC) simulations are often employed for UQ, but they suffer from slow convergence, requiring a large number of samples to achieve accurate statistical estimates. In CFD-related applications, the computational cost of sample evaluation is an important obstacle for UQ and methods that require much less samples than MC are of interest.

An alternative, more efficient approach is Polynomial Chaos Expansions (PCE), which builds a surrogate model to approximate the relationship between uncertain inputs and outputs. The QoI is modeled as a polynomial expansion, with its basis polynomials being functions of the input variables. Similar to other spectral methods, like the Fourier Transform, the goal is the computation of the basis functions' coefficients. In PCE, not only does the acquiring of the coefficients produces a metamodel for the stochastic process of interest, but information about the uncertain QoI's moments can be obtained by merely post-processing the coefficient vector. The foundation of PCE lies in the theory of homogeneous chaos [33], where Hermite polynomials were used as basis functions for random processes with Gaussian input. This framework was later extended to include non-Gaussian cases, [35], based on the generalization of orthogonal polynomials [2]. Within this framework, named Generalized PCE, broader classes of random variables can be accommodated by employing polynomial bases, orthogonal to the joint probability density function (PDF) of the inputs. The method's accuracy was originally demonstrated by applying it to stochastic differential equations and comparing the resulting expectations with analytical solutions [35]. This success motivated their implementation in various UQ problems of different engineering sectors [28]. In a PCE implementation, one must define the polynomial basis (size and type of polynomials) and decide on applying the expansion in an intrusive or non-intrusive manner.

For a given joint PDF of the model input, specific polynomial families are applicable, according to the generalized PCE theory. Once these are obtained, a crucial next step for PCE is the truncation of the polynomial basis. The infinite basis functions are truncated based on the limit of the total order of the multidimensional polynomials. This value is named chaos order.

With the basis defined, the intrusive vs. non-intrusive dilemma occurs. In the intrusive approach, the uncertain quantities of the physical model are replaced by spectral expansions in the polynomial basis. By projecting the governing equations onto this basis and exploiting the orthogonality of the polynomials, the expansion coefficients can be computed through inner product integrals. Applying the intrusive approach to complex models, such as those governed by nonlinear partial differential equations, is significantly challenging. The mathematical reformulation is involved, since the already complex equations must be redefined for the stochastic variables to be introduced. This dictates a tedious procedure of applying fundamental changes to an already complicated source code, such as CFD software. To address these

difficulties, non-intrusive methods were introduced. These are divided into two main categories, projection-based and regression-based PCE [19],[6]. These methods allow for UQ without altering the original evaluation software, making them particularly suitable for CFD and other scientific, simulation-heavy domains. Moreover, the non-intrusive approach allows for the development of PCE-related software, applicable to any deterministic software of choice.

The non-intrusive projection approach, also named Galerkin method, relies on the numerical calculation of the same inner product integrals that intrusive PCE method solves, via quadrature rules. Although this method can guarantee the accuracy of the expansion, its computational requirements render it infeasible in cases of high input dimensionality (many uncertain variables).

The regression-based method (rPCE) is an Ordinary Least Squares (OLS) approach. Samples of the input probability distribution are acquired via Design-of-Experiments (DoE) techniques and their corresponding QoI values are computed. Typical DoE methods include random sampling, Latin Hypercube Sampling (LHS) etc. With the truncation of the polynomial basis and the obtaining of the sample set and its corresponding model evaluations, a linear system of equations can be defined. The basis functions evaluations of all samples yield a design matrix, the model evaluations serve as the right hand side, and the solution vector of the linear system corresponds to the PCE coefficients. For accuracy this system needs to be overdetermined. With sample size being larger than the basis size, there are fewer unknowns than equations. The solution to the oversampled OLS problem is acquired by computing the pseudo-inverse of the design matrix and multiplying it with the right hand side. While rPCE encounters explosive growth in sample requirements in cases with many uncertain variables, these are moderate compared to those of the projection-based method. The characteristics of non-intrusive, rPCE is indeed a suitable choice for CFD-related UQ [18]. For this reason, all PCE applications in this work refer to rPCE

1.3 Sparse PCE

Although less costly than the projection-based method, rPCE still suffers from the curse-of-dimensionality. This is evident in the truncation formula, which describes how the number of basis terms grows rapidly with the increase in the input variables' dimensionality and chaos order. This growth, along with the need for oversampling, results in the requirement of large sample sizes for regression in high-dimensional rPCE, and the need for methods which rely on small experimental sets is still high. In such cases, the sparsity-of-effects principle [20] can be assumed. According to this hypothesis, few terms of the large basis play a significant role in the solution vector. This is explained by the fact that in physical systems, only a limited number of features or low-order interactions significantly affect the output. The true solution can be either sparse, with only a few non-zero terms, or compressible, where few

terms are of significantly larger magnitude and influence than the rest.

Due to the sparsity or compressibility present in high-dimensional rPCE solutions, sparse regression techniques are in use. The crucial property of such regression techniques is that they require fewer samples than unknowns to produce results of reasonable accuracy. Therefore, their successful implementation can yield accurate PCE estimates in undersampled cases, while the OLS approach requires oversampling. This advantage is crucial, due to the significant decrease in the number of required, costly, model evaluations. The use of sparsity in undersampled settings to find sparse solutions is an approach thoroughly documented and explained by the field of Compressed Sensing (CS) [8], whose applications are found primarily in signal processing applications [24]. . The goal of sparse regression is to produce an accurate solution with limited samples, by enforcing sparsity in the solution of an OLS problem, since the omission of unimportant features is not expected to significantly affect the quality of the results. This can be formulated as a constrained optimization problem where the L_0 norm of the solution vector, i.e. the number of its non-zero entries, is minimized and the satisfaction of the Least-Squares equation is the constraint. This problem is not convex due to its combinatorial nature. To surpass this problem, CS literature proposes two strategies: (a) greedy algorithms can select iteratively select a small subset of active terms in the solution and calculate their coefficients, (b) the replacement of the L_0 norm by the L_1 norm (the sum of a vector's absolute values), which is a technique that makes the optimization convex, and still enforces sparsity to the solution due to its interaction with the OLS constraint.

For a greedy implementation, software applying the Orthogonal Matching Pursuit (OMP) algorithm [25] is developed in this diploma thesis. OMP is a greedy algorithm that iteratively selects the basis term that is most correlated with the Least-Squares residual. For L_1 minimization, the popular Least Absolute Shrinkage Operator (LASSO) [29] is adopted in this thesis. LASSO is a linear optimization problem, where both the Least-Squares Residual and the L_1 norm are minimized, with the introduction of a regularization parameter, which acts as a Lagrange operator for the L_1 term. Since the LASSO method requires an optimization algorithm, an optimizer must be defined. An optimization method that yields the solution to the LASSO problem is the Iterative Shrinkage Thresholding Algorithm (ISTA) [4]. This method relies on the proximal gradient method, a mathematical framework designed to handle locally non-differentiable functions, such as the L_1 norm. The algorithm proceeds by computing a gradient descent step on the differentiable component of the objective function (in this case, the Least Squares residual), followed by the application of a soft-thresholding operator. This step acts as the proximal operator for the L_1 norm, effectively incorporating its influence into the update rule and encouraging sparsity in the solution. Software implementing this algorithm is developed and used in this diploma thesis.

The ability of sparse regression to produce accurate solutions with low-size experi-

mental designs has been the motivation behind many successful sparse PCE applications. Notably, sparse PCE has been used in CFD and other simulation-related applications of UQ, [27],[11],[7].

1.4 Robustness of the Regression Algorithm

However, as the size of the chosen sample set decreases, the sensitivity of the PCE result to the quality of the set increases. Samples produced by classical DoE methods, such as random sampling and Latin Hypercube Sampling, include randomness, which affects the result. While randomness is a crucial property for the representation of the sampled distribution, sample sets of the same size can differ in their capability in producing accurate regression results. The capability of the sparse algorithm to provide accurate results is thoroughly explored in the foundation of CS. According to the CS literature, the lower the mutual coherence of the linear system's matrix, i.e. the normalized inner product of its least orthogonal columns, the higher the probability of accurate estimation of the sparse terms [15]. Lowering matrix coherence can ameliorate PCE results in low sample-size, resulting in lowering the necessary computational cost for UQ. This is the motivation behind basis reduction, i.e. the utilization of a limited subset of the potentially large truncated basis set. This reduction can be done a-priori, with hyperbolic truncation [7], or during the regression algorithm. Since the motivation of this study is to compare the accuracy of sparsity-promoting algorithms with those of the OLS in similar conditions, hyperbolic truncation is not employed. While basis reduction is a fundamental part of OMP, its greedy nature, while providing computational efficiency, can limit its capability in identifying the sparse solution [31]. For this reason, a method where the reduction of basis size will be done via an impartial screening of the full basis set is desired. The standard LASSO method does not introduce any basis reduction and considers all the given basis terms. Its regularization parameter is tuned via Cross-Validation so the activated terms emerge from the data without the need of either a-priori assumptions or greedy processes. Consequently, a two-stage LASSO approach is implemented where the first LASSO run simply serves as a basis activation step, where the basis terms with non-zero coefficients are members of the active set. The basis of the second run contains only the active set from the first run. With significantly fewer features and lower coherence, the second regression can estimate the important terms with greater accuracy. Different approaches that use LASSO for basis reduction exist, using OLS on the active set (post-LASSO) [5], or a second LASSO with smaller regularization parameters (relaxed LASSO) [22].

1.5 Thesis Objective and Layout

The objectives of this work are:

- to develop and implement Orthogonal Matching Pursuit (OMP) and LASSO

as representative sparsity-promoting regression algorithms,

- to extend the analysis with reduced-basis strategies, namely post-LASSO and relaxed LASSO,
- to compare these sparse and reduced-basis methods against OLS in terms of required sample evaluations,
- and to assess their ability to construct accurate UQ surrogates for aerodynamics-related applications under limited sample availability.

The remainder of this thesis is structured to develop the theoretical framework of sparse PCE and demonstrate their application to aerodynamic UQ through progressively more complex applications. Chapter 2 introduces the theoretical background necessary for understanding rPCE and sparse regression. It covers the construction of orthogonal polynomial bases, methods for sampling and coefficient estimation, and the statistical interpretation of the resulting expansions. The sample requirements of rPCE motivate the use of sparsity-promoted techniques. Sparse regression on undersampled linear systems is thoroughly examined by the field of CS, and the two main approaches consist of either greedy algorithms (such as OMP) and L_1 minimization (such as LASSO). Crucial for robust sparse solutions is the mutual coherence of the regression matrix, with low coherence dictating high probability for an accurate sparse result. For PCE-related sparse regression, it is demonstrated that the regression matrix in high-dimensional settings suffers from high mutual coherence, which degrades the quality of sparse regression. This issue motivates the need for basis reduction strategies. For this reason, different sparse regression techniques, which involve basis reduction are discussed.

Chapter 3 presents the numerical validation of the aforementioned methods through three applications. The first is the Borehole function [17], a low cost benchmark in which the different sparse algorithms can be compared in their ability in undetermined rPCE. Initial conclusions are drawn from this analysis and the comparison of the sparse regression methods is repeated in the aerodynamics-related cases that follow. The second case applies sparse PCE to the simulation of the NACA 2412 airfoil using the XFOIL [12] solver under uncertain flow conditions. The third case involves a CFD simulation [3] of the NACA 16103 airfoil, incorporating geometric uncertainties modeled via the Karhunen-Loève Expansion (KLE). [32]. Chapter 4 summarizes the main conclusions of the thesis, highlighting the trade-offs between accuracy, sparsity, computational efficiency, and discussing possible directions for future work.

Chapter 2

Theoretical Background

2.1 PCE

PCE [35] is a way to represent the response of a system as a series expansion in terms of polynomials orthogonal to the probability distribution of input uncertainties. This concept, originally introduced as the Wiener-Hermite expansion, has evolved into a powerful meta-modeling technique for UQ. The key advantage this method holds over a MC simulation is its ability to compute the statistical moments of the QoI distribution with a significantly lower number of samples.

2.1.1 Homogeneous Chaos and Basis Selection

PCE is based on the theory of *homogeneous chaos* [33], where a Gaussian process is represented using a basis of Hermite polynomials, which in turn are orthonormal to the Gaussian PDF. This was later generalized for other cases of input distributions, via generalized PCEs [35]. The expansion is given by:

$$u(\boldsymbol{\xi}) = \sum_{\alpha=0}^{P-1} c_\alpha \, \Psi_\alpha(\boldsymbol{\xi}), \tag{2.1}$$

where: - $u(\xi)$ is the QoI,

- ξ represents the vector of random input variables,

- $\Psi_\alpha(\xi)$ are the multivariate orthonormal polynomials,

- P is the number of basis functions,

- c_α are the expansion coefficients, which must be estimated.

The orthonormality of the basis polynomials $\Psi_\alpha(\xi)$ is defined with respect to the joint PDF $\rho(\xi)$ of the input variables. This condition must hold as follows:

$$\int_{\mathbb{R}^d} \Psi_\alpha(\xi)\Psi_\beta(\xi)\rho(\xi)\, d\xi = \delta_{\alpha,\beta}, \tag{2.2}$$

where: - $\delta_{\alpha,\beta}$ is the Kronecker symbol, which equals 1 when $\alpha = \beta$ and 0 otherwise, - $\rho(\xi)$ is the joint PDF of the input variables ξ.

Eq. (2.2) ensures that the expansion is capable of converging to an accurate solution. It is important to note that orthogonality is the necessary condition, and the method can converge even if the norms of the polynomials are not unitary. However, the normalization of the basis functions facilitates important calculations, which will be seen below. In conclusion, the choice of polynomials corresponds to the distribution of the input variables. For example, Legendre and Hermite polynomials are the associated orthogonal polynomial families for uniform and Gaussian distributions, respectively. These polynomials can be modified for their norms to be unitary, which makes the moment extraction more straightforward, as it will be seen later.

2.1.2 Normalized Hermite Polynomials for Gaussian Distributions

For the Gaussian distribution with mean $\mu = 0$ and variance $\sigma^2 = 1$, the orthonormal polynomials are the *Hermite polynomials*, $H_n(x)$, defined as follows:

$$H_n(x) = (-1)^n e^{x^2} \frac{d^n}{dx^n}\left(e^{-x^2}\right).$$

The normalized Hermite polynomials are given by:

$$\psi_n(x) = \frac{1}{\sqrt{2^n\, n!\, \sqrt{\pi}}}\, H_n(x)\, e^{-x^2/2} \tag{2.3}$$

These polynomials are orthogonal with respect to the Gaussian PDF $\rho(x)$:

$$\rho(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}.$$

The orthonormality condition holds:

$$\int_{-\infty}^{\infty} \psi_m(x)\psi_n(x)\rho(x)\, dx = \delta_{m,n}.$$

By using the appropriate orthonormal polynomial families, such as Hermite polynomials for Gaussian distributions, the PCE provides a structured and convergent way to represent the uncertain QoI, regardless of the underlying complexity of the

relationships between input variables and the QoI. The generalized PCE framework [35] describes the different types of polynomials that correspond to different types of input PDFs (Table 2.1).

Table 2.1: *Orthogonal polynomials used in PCE for selected input probability distributions.*

Distribution	**Support**	**Orthogonal Polynomial**	**PDF**
Uniform $\mathcal{U}[a,b]$	$[a,b]$	Legendre	Constant Weight
Gaussian $\mathcal{N}(0,1)$	$(-\infty,\infty)$	Hermite	e^{-x^2}
Gamma $\Gamma(\alpha,\beta)$	$[0,\infty)$	Laguerre	$x^{\alpha-1}e^{-x}$
Beta Beta(α,β)	$[0,1]$	Jacobi	$x^{\alpha-1}(1-x)^{\beta-1}$
Exponential	$[0,\infty)$	Laguerre	$\lambda e^{-\lambda x}$ Gamma with $\alpha=1$

2.1.3 Total-Order Truncation of Polynomial Basis

Before constructing the PCE surrogate, the polynomial basis must be truncated to a finite set of multivariate polynomials. A common choice is total-order truncation [35], in which all multivariate polynomial terms up to a fixed total degree p across d uncertain variables are retained. This truncation scheme is named total order, to differentiate it from methods that keep a subset of those variables, like hyperbolic truncation [7]. It must be noted that the truncation of the polynomial basis is necessary for all types of PCE approaches and sparsity, which will be discussed later refers to the activation of a fraction of the already truncated basis. The size of this basis is given explicitly by the combinatorial formula:

$$P = \binom{d+p}{p} = \frac{(d+p)!}{d!\,p!} \tag{2.4}$$

This value determines the number of terms in the expansion, the dimension of the coefficient vector $\mathbf{c}$, and the minimal number of model evaluations required by regression-based methods. It also governs the quadrature growth rate in projection-based approaches.

2.1.4 Moment Extraction

The most important information that an expansion provides are the statistical moments of the QoI distribution. Simple post-processing of the expansion's coefficients can directly compute the first two moments.

1. Mean

The mean is the expectation of u, given by:

$$\mathbb{E}[u] = \mathbb{E}\left[\sum_{\alpha} c_\alpha \Psi_\alpha(\xi)\right] = \sum_{\alpha} c_\alpha \mathbb{E}[\Psi_\alpha(\xi)] = \sum_{\alpha} c_\alpha \int_\Omega \Psi_\alpha(\xi)\, \mathrm{d}\mathbb{P}(\xi).$$

By orthonormality of the basis functions with respect to the probability measure $\mathbb{P}$:

$$\int_\Omega \Psi_\alpha(\xi)\, \mathrm{d}\mathbb{P}(\xi) = \int_\Omega \Psi_\alpha(\xi)\Psi_0(\xi)\, \mathrm{d}\mathbb{P}(\xi) = \delta_{\alpha 0}, \quad \text{since } \Psi_0(\xi) = 1.$$

Thus, the mean is:

$$\mathbb{E}[u] = c_0 \underbrace{\int_\Omega \Psi_0(\xi)\, \mathrm{d}\mathbb{P}(\xi)}_{=1} + \sum_{\alpha \neq 0} c_\alpha \underbrace{\int_\Omega \Psi_\alpha(\xi)\, \mathrm{d}\mathbb{P}(\xi)}_{=0} = \boxed{c_0}.$$

2. Variance and Standard Deviation

The variance is given by:

$$\mathrm{Var}(u) = \mathbb{E}[(u - \mathbb{E}[u])^2] = \mathbb{E}\left[\left(\sum_{\alpha \neq 0} c_\alpha \Psi_\alpha(\xi)\right)^2\right] = \int_\Omega \left(\sum_{\alpha \neq 0} c_\alpha \Psi_\alpha(\xi)\right)^2 \mathrm{d}\mathbb{P}(\xi).$$

Expanding the square:

$$\mathrm{Var}(u) = \sum_{\alpha \neq 0} \sum_{\beta \neq 0} c_\alpha c_\beta \int_\Omega \Psi_\alpha(\xi)\Psi_\beta(\xi)\, \mathrm{d}\mathbb{P}(\xi).$$

By orthonormality, $\int_\Omega \Psi_\alpha(\xi)\Psi_\beta(\xi)\, \mathrm{d}\mathbb{P}(\xi) = \delta_{\alpha\beta}$. Therefore:

$$\mathrm{Var}(u) = \sum_{\alpha \neq 0} c_\alpha^2 \underbrace{\int_\Omega \Psi_\alpha^2(\xi)\, \mathrm{d}\mathbb{P}(\xi)}_{=1} = \boxed{\sum_{\alpha \neq 0} c_\alpha^2}.$$

For the Standard Deviation (SD):

$$\sigma_u = \sqrt{\mathrm{Var}(u)} = \sqrt{\sum_{\alpha \neq 0} c_\alpha^2}. \tag{2.5}$$

It is evident how the unitary norm of the basis polynomials facilitates the moment extraction since there is no need to divide the expectation integrals with the polynomials' norm.

2.1.5 Non-Intrusive PCE

In this work, rPCE is used. For the choice of this method to be understood, the difference between intrusive and non-intrusive approaches as well as projection and regression-based expansions must be examined.

Intrusive methods [9] involve modifying the underlying governing equations of the model. Expressions, such as equation (2.1), replace the uncertain variables and the unknown functions in the model equations. This method, while used in the first applications of the PCE method, mainly in stochastic ordinary differential equations, can become really challenging when dealing with engineering problems. Not only is it mathematically more challenging to introduce these expansions in complicated systems of partial differential equations, but fundamental changes to usually already complex software need to be implemented.

On the other hand, non-intrusive approaches [14], treat the model as a black box and require simulations of the original deterministic model for the computation of the PCE coefficients. The black box use of the deterministic model is in the form of necessary evaluations of experimental sets in the non-intrusive rPCE, or collocation points for the non-intrusive Galerkin projection approach. Non intrusive PCE allows for the development and implementation of general use PCE software, regardless of the physical problem in which UQ is performed.

2.1.6 Galerkin-Projection vs. Regression

In non-intrusive PCE, the two main approaches for computing the coefficients of the expansion are Galerkin projection and regression.

In **Galerkin projection**, [13], [34], these coefficients are computed by applying projection integrals to equation 2.1:

$$c_i = \frac{\langle u, \Psi_i \rangle}{\langle \Psi_i, \Psi_i \rangle} = \frac{\int u(\boldsymbol{\xi}) \Psi_i(\boldsymbol{\xi}) \rho(\boldsymbol{\xi})\, d\boldsymbol{\xi}}{\int \Psi_i^2(\boldsymbol{\xi}) \rho(\boldsymbol{\xi})\, d\boldsymbol{\xi}},$$

where $\rho(\boldsymbol{\xi})$ is the joint PDF. The integral in the denominator is known, equal to the norm of the basis functions, whereas that in the numerator must be computed. It is approximated using a numerical quadrature rule:

$$c_{\boldsymbol{\nu}} \approx \sum_{j=1}^{N_q} w^{(j)} u(\boldsymbol{\xi}^{(j)}) \Psi_{\boldsymbol{\nu}}(\boldsymbol{\xi}^{(j)}),$$

where:

- $\boldsymbol{\xi}^{(j)}$, $j = 1, \ldots, N_q$, are the quadrature nodes,
- $w^{(j)}$, $j = 1, \ldots, N_q$, are the corresponding quadrature weights,

- N_q is the total number of quadrature points.

Size of the Quadrature Grid To exactly integrate all basis functions up to total degree $2p$, the quadrature rule must have a degree of precision at least $2p$. The required number of quadrature nodes depends on the dimensionality d and the type of quadrature used:

- For **tensor product Gauss quadrature** (exact in 1D for polynomials of degree up to $2p-1$), the number of nodes grows exponentially:

$$N_q = (p+1)^d.$$

- **Sparse grid quadrature** (e.g., Smolyak) can be used in PCE [34] which results in the more moderate growth of required nodes. Not to be confused with the sparse PCEs, sparsity in this context means that low-order terms are kept from the large tensor product of univariate quadrature rules. While sparse grids significantly reduce computational requirements, there still is exponential growth with dimensionality, and their use can be practically infeasible in cases with a large number of input variables and considerable computational cost of model evaluation.

The rPCE method [6] avoids quadrature by casting the coefficient estimation as a least-squares problem. Given N samples $\{\boldsymbol{\xi}^{(j)}, u^{(j)}\}_{j=1}^{N}$, the system is written as:

$$\mathbf{A}\mathbf{c} \approx \mathbf{u},$$

where $\mathbf{A} \in \mathbb{R}^{N \times P}$ with entries $A_{ji} = \Psi_i(\boldsymbol{\xi}^{(j)})$, $\mathbf{c} \in \mathbb{R}^P$ is the vector of unknown coefficients, and $\mathbf{u} \in \mathbb{R}^N$ contains the model outputs. In cases with costly simulation software, such as CFD, the computation of u takes up the largest portion of computational cost, which is why the sample size N is limited to the computational capabilities of the application. The least-squares solution minimizes $\|\mathbf{A}\mathbf{c} - \mathbf{u}\|_2^2$. This approach is more computationally feasible for high-dimensional problems, but it requires oversampling (i.e., $N > P$) to ensure that the problem is well-conditioned, with the optimal oversampling ratio depending on the smoothness of the response, sparsity of the true coefficients, and noise levels in the data.

In summary, Galerkin projection offers higher theoretical accuracy but is computationally intensive in high dimensions due to evaluation costs. The rPCE approach is more flexible and better suited for high-dimensional UQ, provided that careful attention is paid to the sampling strategy and oversampling level.

2.1.7 rPCE Methodology

The rPCE methodology is an OLS approach, organized into four main steps: (1) definition and truncation of the polynomial basis, (2) DoE and construction of the

linear system, where samples are defined and evaluated and the largest portion of the method's computational cost is found (3) coefficient estimation via singular value decomposition (SVD)–based pseudoinverse, and (4) extraction of statistical moments.

Basis Definition and Truncation

A multivariate orthogonal polynomial basis $\{\Psi_i(\boldsymbol{\xi})\}_{i=0}^{P-1}$ is selected (Table 2.1), of total degree up to p using truncation. The total number of terms is explicitly given by eq. (2.4).

DoE

To construct a well-conditioned regression system, N samples $\{\boldsymbol{\xi}^{(j)}\}_{j=1}^{N}$ are generated in the d-dimensional input domain, using a sampling ratio $SR > 1$:

$$N = SR \times P, \qquad SR \approx 1.5\text{–}5,$$

where P is the total number of basis terms. Oversampling ensures accuracy of the least-squares fit by controlling the condition number of the design matrix.

Common sampling methods include random sampling and LHS [21]. These methods produce N samples of input dimensionality with the sample size being a user-defined parameter. Both of these methods require a random-number-generator (RNG), i.e. a method that produces random (or pseudo-random) numbers and the inverse cumulative distribution function related to the multivariate distribution of the input.

Random Sampling: With the help of RNG, N samples on the unit cube $[0,1]^d$ are generated.

LH Sampling: The unit interval $[0,1]$ is divided into N equal-probability strata for each marginal. Using the RNG, one uniform sample $u_k^{(j)}$ is drawn from each stratum, then these samples across strata are randomly permuted to form $\mathbf{u}^{(j)} \in [0,1]^d$, adding random behavior to the sampling algorithm.

Since both LHS and random sampling produce samples in $[0,1]^d$, each coordinate is mapped to its true marginal via inverse transform sampling:

$$\xi_k^{(j)} = F_k^{-1}\big(u_k^{(j)}\big),$$

where F_k^{-1} is the inverse cumulative distribution function of ξ_k. This is possible because of the independence of the random variables, and the joint cumulative density function is just the product of the individual components. This preserves the uniformity or low-discrepancy structure in the target probability space, provided

the cumulative density functions are continuous and monotonic. In this work, LHS is preferred, due to its space-filling capabilities in cases of high dimensionality.

Each basis function is evaluated at the N samples to build the design matrix $\mathbf{A} \in \mathbb{R}^{N \times P}$ with entries:

$$A_{ji} = \Psi_i\left(\boldsymbol{\xi}^{(j)}\right),$$

and the model outputs of these samples are collected into $\mathbf{u} = [u^{(1)}, \ldots, u^{(N)}]^T$.

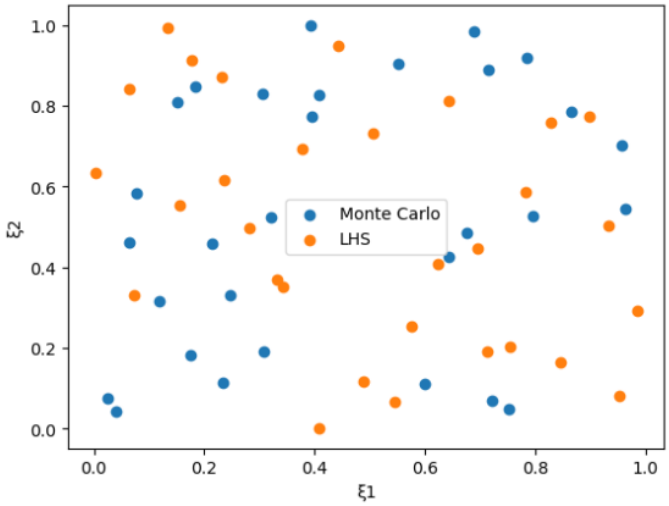

Figure 2.1: *Random (blue) and LH (orange) samples in a 2D input space of Uniform variables.*

SVD for the solution of OLS

With the linear system defined, $\mathbf{Ac} = \mathbf{u}$, the OLS approach is completed with its solution. The overdetermined system is solved for the coefficient vector $\mathbf{c} \in \mathbb{R}^P$ using the Moore-Penrose pseudoinverse. While the pseudoinverse can be obtained through the normal equations, $\mathbf{c} = (\mathbf{A}^\top \mathbf{A})^{-1}\mathbf{A}^\top \mathbf{u}$, the most popular method is Singular Value Decomposition (SVD), due to its robustness to ill-conditioned matrices. After performing the SVD on the regression matrix:

$$\mathbf{A} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$$

The pseudoinverse can be calculated:

$$\mathbf{A}^\dagger = \mathbf{V}\boldsymbol{\Sigma}^\dagger\mathbf{U}^T$$

With the pseudoinverse, the coefficient vector can be computed:

$$\mathbf{c} = \mathbf{A}^\dagger\mathbf{u}.$$

2.2 Motivation for Sparse PCE

As described above, the number of needed samples is proportional to the number of unknowns (oversampling factor) in rPCE, which depends on the number of uncertain variables and chaos order, as demonstrated in eq. (2.4). High-dimensional problems can lead to a prohibitively large necessary sample size. The steep growth in basis terms (Fig. 2.2) is a case of Curse of Dimensionality. The large number of terms implies that potentially, only a small portion of them will play a significant role in the PCE solution. The justification for sparsity arises from the *sparsity-of-effects principle* [20], which suggests that physical systems are primarily governed by a few dominant variables and low-order interactions. Even when exact sparsity is not present, many models are *compressible*, i.e. their PCE coefficients tend to decay rapidly in magnitude when sorted, allowing the system behavior to be captured effectively using only a few terms. Hence, if few terms are important, it is desirable to take advantage of sparsity or compressibility of PCEs, in order to reduce the required sample size. Sparse regression and CS techniques can be utilized to achieve this goal. Their correct use results in the activation and calculation of the few important terms in undersampled settings. To sum up, in high-dimensional regression problems such as rPCE, it is reasonable to assume that only a subset of basis terms carries most of the information. Exploiting this sparsity reduces the number of required evaluations and thus lowers the overall computational cost.

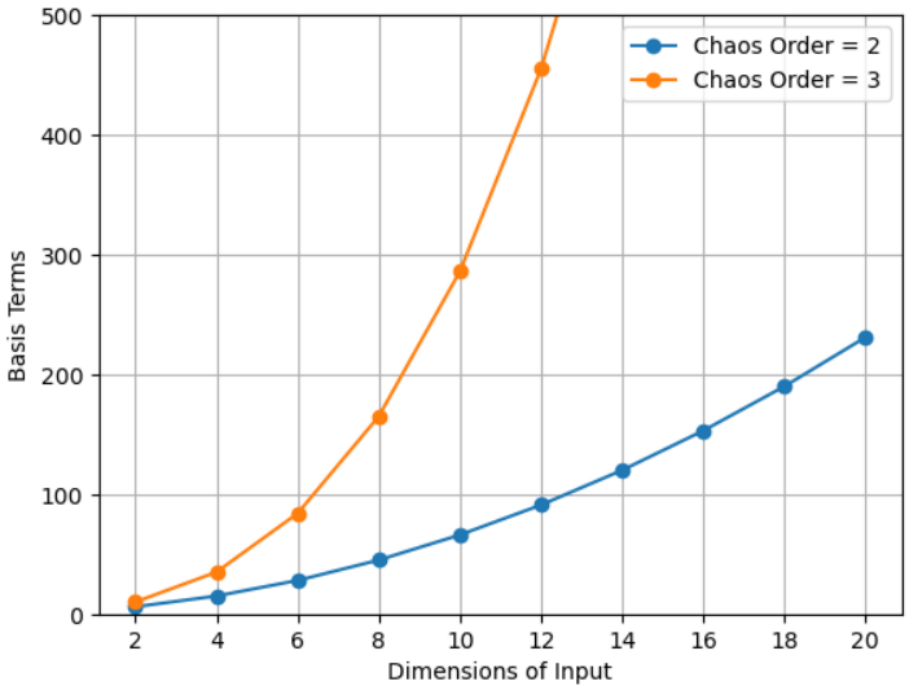

Figure 2.2: *Growth of Basis Terms with respect to dimensionality, with chaos order $p = 2$ (blue line) and $p = 3$ (orange line).*

The following chapter explores how CS theory and algorithms can be employed to construct sparse PCEs efficiently and reliably.

2.3 CS–Inspired Sparse PCE

CS [8] is a revolutionary signal processing technique that allows for the reconstruction of sparse signals from a small number of measurements. This technique leverages the fact that many natural solutions are sparse or compressible in some basis, hence they can be represented with only a few non-zero coefficients. The foundational work in CS [8] demonstrated that if a vector is sparse in some basis (e.g. Fourier coefficients), it can be recovered from far fewer samples than traditionally required (e.g. by the Nyquist-Shannon sampling theorem). This can also be implemented in rPCE applications of high-dimensionality, where the coefficient vector is expected to be sparse in the truncated polynomial basis, in order to limit sample size requirements. For this reason, apart from applications in signal processing-related fields, including medical imaging, radar, and wireless communications [24], CS approaches have been recently implemented in rPCE [20], including CFD-related applications [27].

2.3.1 Sparsity and Its Importance

A vector $\mathbf{c} \in \mathbb{R}^P$ is said to be k-sparse if it contains only k non-zero entries, where $k \ll N$. Of interest are sparse solutions to linear systems:

$$\mathbf{A}\mathbf{c} = \mathbf{u},$$

where:

- $\mathbf{c}$ is a sparse vector with $\|\mathbf{c}\|_0 = k$

- $\|\mathbf{c}\|_0$ (L_0 norm) denotes the number of non-zero elements in $\mathbf{c}$,

- $\mathbf{A} \in \mathbb{R}^{N \times P}$

- $\mathbf{u} \in \mathbb{R}^N$

In rPCE, we expect the model response $\mathbf{u}$ to have a sparse solution $\mathbf{c}$ in the polynomial basis $\mathbf{\Psi}$, due to the importance of few terms. The importance of sparsity lies in its ability to enable the recovery of solutions to undersampled problems. In traditional linear algebra, solving an under-determined system $\mathbf{A}\mathbf{c} = \mathbf{u}$, with $N < P$, is not possible, because there are infinitely many solutions. However, if $\mathbf{c}$ is known to be sparse, it is possible to recover it accurately using sparsity-promoting techniques, provided that the number of active terms is small compared to the basis terms ($k \ll N < P$). The key goal of the sparsity-promoting algorithms is to modify the OLS approach, in order to activate a few dominant terms. Hence, not only should

the solution be sparse, but the algorithm must correctly identify the significant basis terms.

2.3.2 L_0 Minimization

The L_0 norm of a vector is the number of its non-zero entries, and therefore an optimization problem with the goal of recovering the sparse solution of a Least-Squares problem is the following:

$$\min_{\mathbf{c}} \|\mathbf{c}\|_0 \quad \text{subject to} \quad \mathbf{Ac} = \mathbf{u} \tag{2.6}$$

However, given that a k-sparse vector is the true solution to the system, the solution to this optimization problem could fail to identify it. Firstly, for the L_0 minimization to be properly defined, the number of active terms in the sparse solution needs to be smaller than the number of matrix rows in eq. (2.6) ($k \ll N$). If this is not true, a solution to the problem comes from arbitrarily selecting N basis terms and omitting the rest, the matrix may be inverted and a solution with zero (or practically zero) Least-Squares residual is obtained. This would serve as a solution since the Least-Squares constraint would be satisfied and the minimum L_0 would be achieved. While this solution would satisfy the optimization conditions, it would be of no actual predictive value, as it would be a Least-Squares fit of an arbitrarily selected basis subset. Hence, for the true sparse solution to be searched for, it is essential that the L_0 norm of the true sparse solution is smaller than the number of rows. Secondly, handling an expected Least-Squares error is crucial. In undersampled rPCE, sources of error exist and influence the outcome of the sparse optimization problem. The desired sparse solution would exhibit a small, non-zero Least-Squares error due to the truncation of basis terms and the avoidance of overfitting to a small dataset. If the Least-Squares constraint is an equality, this effect will not be accurately modeled, and the arbitrary solutions described earlier would still be preferred. The original sparsity-promoting optimization problem needs to be redefined:

$$\min_{\mathbf{c}} \|\mathbf{c}\|_0 \quad \text{subject to} \quad \|\mathbf{Ac} - \mathbf{u}\|_2 < \epsilon \tag{2.7}$$

where the small value of ϵ is an important hyperparameter that needs to be tuned. Both problems of eq. (2.6) and eq. (2.7) are non-convex due to the combinatorial nature of the L_0 norm. To surpass this problem, there are two fundamental ways, proposed by CS literature: (a) Greedy algorithms [31], such as OMP, (b) L_1 instead of L_0 minimization with methods such as Basis Pursuit [10] and the LASSO [29].

2.3.3 The OMP algorithm

Orthogonal Matching Pursuit (OMP) [25] is a greedy algorithm for sparse regression that builds the support, i.e. the active basis terms, of $\mathbf{c}$ one index at a time, using

the correlation of a basis with the current residual, which is their inner product. This method's greedy nature qualifies it as an optimizer to the nonconvex sparse problem eq. (2.7). At each iteration, the algorithm: (a) selects the column of $\mathbf{A}$ most correlated with the current residual, (b) augments the active set, (c) computes the best least-squares fit on that support, and (d) updates the residual. Termination occurs when the residual reaches a certain tolerance, or a user-defined number of basis terms is included.

The procedure in detail:

1. **Initialize:** residual $\mathbf{r}^{(0)} = \mathbf{u}$, support $S^{(0)} = \varnothing$, iteration $j = 0$, coefficients $\mathbf{c}^{(0)} = \mathbf{0}$.

   In the start of the algorithm, the solution vector is an array of zeros, the active basis set (the terms whose coefficients are non-zero) is empty and the residual is equal to the right hand side. The algorithm will terminate if a user-defined number of terms is selected, or prior to that the Least-Squares residual drops below a fixed tolerance.

2. **Loop:** while $|S^{(j)}| < k$ *and* $\|\mathbf{r}^{(j)}\|_2 > \epsilon$, do

   (a) *Atom selection (greedy step):*

   $$\mathbf{i^{(j+1)}} \;=\; \arg\max_{i\in\{1,\dots,P\}} \left|\mathbf{a}_i^\top \mathbf{r}^{(j)}\right| \quad \text{(if columns are not normalized, use } |\mathbf{a}_i^\top \mathbf{r}^{(j)}|/\|\mathbf{a}_i\|_2).$$

   Update the support: $\mathbf{S^{(j+1)}} = S^{(j)} \cup \{\, i^{(j+1)}\}$.

   In the atom selection step, the column of the regression matrix that exhibits the highest correlation with the current residual is selected, and the active basis set is augmented with this term. This greedy strategy is inherently path-dependent: at each iteration, the choice of the next basis function relies solely on the present residual, which is itself determined by all previous selections

   (b) *Coefficient update (restricted least squares):*

   $$\mathbf{c}^{(j+1)}_{S^{(j+1)}} \;=\; \arg\min_{\mathbf{c}} \left\|\mathbf{A}_{S^{(j+1)}}\mathbf{c} - \mathbf{u}\right\|_2^2, \qquad \mathbf{c}^{(j+1)}_{[P]\setminus S^{(j+1)}} \;=\; \mathbf{0}.$$

   With the basis set defined, the coefficients are updated via simple OLS (which can be done via SVD).

   (c) *Residual update:*

   $$\mathbf{r}^{(j+1)} \;=\; \mathbf{u} - \mathbf{A}\,\mathbf{c}^{(j+1)}.$$

3. **Return:** $\mathbf{c} \leftarrow \mathbf{c}^{(j)}$ (and the support $S^{(j)}$).

   The new coefficients yield a new residual for the loop to continue.

where:

- $\mathbf{A} \in \mathbb{R}^{N \times P}$ is the *regression/design matrix*; its ith column (atom) is $\mathbf{a}_i$.
- $[P] = \{1, \ldots, P\}$ is the column index set.
- $S \subseteq [P]$ is the *support* (selected indices); $\mathbf{A}_S$ is the submatrix of $\mathbf{A}$ with columns in S.
- For a vector $\mathbf{c} \in \mathbb{R}^P$, $\mathbf{c}_S$ (resp. $\mathbf{x}_{S^c}$) denotes $\mathbf{c}$ restricted to indices in S (resp. its complement).
- $\mathbf{u} \in \mathbb{R}^N$ are the observations/data; $\mathbf{c} \in \mathbb{R}^P$ are the coefficients to estimate.
- $\mathbf{r}$ is the residual $\mathbf{u} - \mathbf{A}\mathbf{x}$; $\|\cdot\|_2$ is the Euclidean norm; $(\cdot)^\top$ denotes transpose.
- k is the desired sparsity level; ϵ is the stopping tolerance on the residual norm.

2.3.4 L_1 Minimization and the LASSO

The use of the L_1 instead of the L_0 norm is also predominant in CS literature, due to the convexity of the L_1 minimization problem. The L_1 norm $\|\mathbf{c}\|_1$ is defined as the sum of the absolute values of the entries of $\mathbf{c}$:

$$\|\mathbf{c}\|_1 = \sum_{i=1}^{n} |c_i|.$$

The L_1 minimization problem, similar to eq. (2.6) is defined:

$$\min_{\mathbf{c}} \|\mathbf{c}\|_1 \quad \text{subject to} \quad \mathbf{A}\mathbf{c} = \mathbf{u}$$

This optimization problem is formally named Basis Pursuit (BP) [10]. As was the case for the L_0 minimization, this problem can be modified for real-world scenarios where Least-Squares error is expected:

$$\min_{\mathbf{c}} \|\mathbf{c}\|_1 \quad \text{subject to} \quad \|\mathbf{A}\mathbf{c} - \mathbf{u}\|_2 < \epsilon \tag{2.8}$$

This is the error-aware variant of BP, formally named Basis-Pursuit-Denoising (BPDN).

With the L_0 norm being a direct metric for sparsity, it is important that its replacement by the L_1 norm still promotes sparse solutions. The L_1 norm may not measure sparsity, but its minimization promotes sparse solutions in undersampled linear regression problems, due to its interaction with linear constraints. While this phenomenon is theoretically explained in CS literature ([8],[10]), it can also be illustrated by the geometry of the L_1 ball, i.e. the shape of iso-L_1 values in a simple two-dimensional example (Fig. 2.3). In two dimensions, the L_1 ball is diamond-shaped, with lines of opposite slopes in the different quadrants, resulting in sharp

corners on the coordinate axes. As demonstrated in Fig. 2.3, points that satisfy linear constraints (represented by a line in the two-dimensional space) obtain the minimum L_1 value in their intersection with these corners, where one of the dimensions is inactive. In practice, the number of dimensions for cases with sparse solutions is large and visualization of this effect is impossible, but this example serves as a simple demonstration of how the L_1 promotes sparse solutions to linear problems.

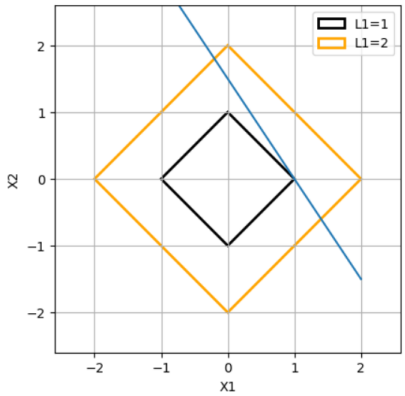

Figure 2.3: *The minimum L_1 (black rhombus), with $L_1 = 1$ and linear constraint (blue line) intersect on a point where x_2 is 0. The sharp corners of the L_1 ball produce sparse solutions to linear problems. The $L_1 = 2$ shape (orange rhombus), is illustrated to show how the iso-L_1 shape is scaled.*

The LASSO optimization problem

The constrained L_1 problem of BPDN (eq. (2.8)) can be reformulated with the use of Lagrange multipliers. More specifically, The LASSO problem [29] is an unconstrained alternative to the BPDN Algorithm, with a Lagrange multiplier term of the L_1 norm. The LASSO problem is formulated as:

$$\min_{\mathbf{c}} \frac{1}{2} \left\| \mathbf{Ac} - \mathbf{u} \right\|_2^2 + \lambda \left\| \mathbf{c} \right\|_1 \tag{2.9}$$

where λ is a regularization parameter that controls the trade-off between the quality of the Least - Squares fit and the sparsity of the solution. In this unconstrained formulation, it is crucial that the regularization parameter is tuned, obtaining a role similar to the Least-Squares residual in BPDN (ϵ, in eq. (2.8)). A common strategy for hyperparameter tuning in regression problems is cross-validation.

In this work, the LASSO approach was preferred for L_1 minimization since its unconstrained nature facilitates the optimization procedure. The LASSO optimization problem is solved with the use of the Iterative Shrinkage-Thresholding Algorithm (ISTA) as an optimizer. ISTA is a simple algorithm for solving L_1-regularized optimization problems, based on gradient-descent and proximal operators [4].

ISTA - LASSO Optimizer

The ISTA [4] is a first-order method tailored to efficiently solve the LASSO problem (eq. (2.9)). The important aspect of ISTA is its ability to handle the non-differentiability of the L_1 norm at 0. The theoretical foundation behind ISTA is the proximal operator [23], a mathematical framework for objective functions that include the Least-Squares function and a non-differentiable convex term, where classical gradient descent cannot be defined. The proximal operator is the solution to a quadratic subproblem, whose minimum is in the direction of the original's objective optimum. ISTA can find the proximal operator for the LASSO problem by decomposing the objective into a smooth part, $\mathbf{f(c)} = \frac{\mathbf{1}}{\mathbf{2}}\|\mathbf{Ac} - \mathbf{u}\|_{\mathbf{2}}^{\mathbf{2}}$, and a non-smooth part, $\lambda\|\mathbf{c}\|_{\mathbf{1}}$.

ISTA proceeds with two main steps in each iteration j:

1. Gradient Descent Step: The smooth part $f(c)$ is minimized via a steepest descent with a fixed step t:

$$\mathbf{z^{(j)}} = c^{(j)} - t\nabla f\left(c^{(j)}\right),$$

where the gradient is given by

$$\nabla\mathbf{f(c)} = A^{\top}(Ac - u).$$

The fixed step size t is critical for ensuring convergence. It must satisfy

$$\mathbf{t} \leq \frac{1}{L},$$

with L being the square of the spectral norm of A, i.e. its largest singular value, (which can be found via SVD).

2. Soft-Thresholding Step: To enforce sparsity, a soft-thresholding operator is applied to the Least-Squares gradient-descent update $z^{(j)}$:

$$\mathbf{c^{(j+1)}} = \mathcal{S}_{\lambda t}\left(z^{(j)}\right),$$

where the soft-thresholding operator is defined element-wise as

$$\mathcal{S}_{\lambda\mathbf{t}}(\mathbf{z_i}) = \text{sign}(z_i)\max\{|z_i| - \lambda t, 0\}.$$

3. Termination: These steps are repeated until the change in c between successive iterations is below a predetermined tolerance:

$$|\mathbf{c^{(j+1)}} - \mathbf{c^{(j)}}|_2 <= \epsilon,$$

Negligible change in c (with ϵ being close to 0) means that the proximal operator has approached the minima of the optimization problem enough that the output can be considered the solution of the LASSO.

Fig. 2.4 illustrates the effect of the soft-thresholding operator on the gradient descent step. For a component of the solution vector, if the absolute value of the gradient descent update is larger than the threshold λt it is simply shrunk towards 0 by λt (two opposite slope lines in Fig. 2.4) , which is the gradient update of the L_1 outside 0. If not, then for the current optimization step, this component is not important enough for the Least-Squares problem to survive the L_1 penalty. Components that do not manage to escape this range are attributed the value of 0, and this is how this optimizer yields sparse solutions.

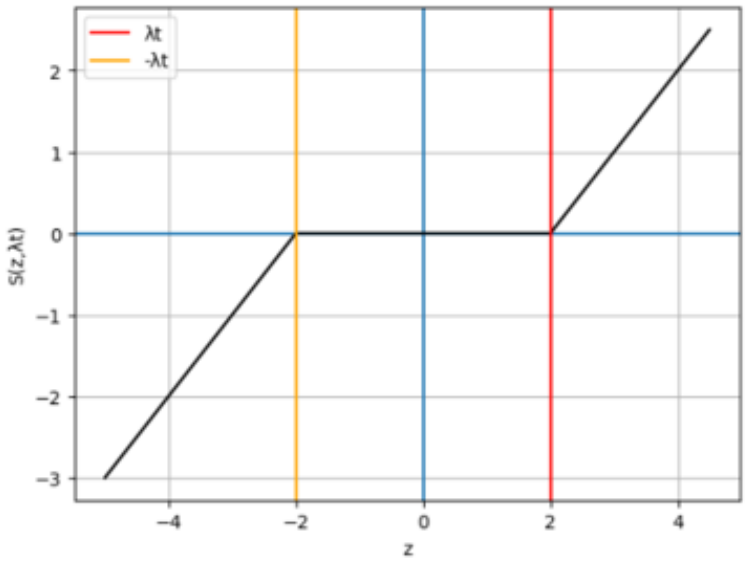

Figure 2.4: *Soft-thresholding $\mathcal{S}_{\lambda t}(\cdot)$ (black line) on the gradient descent update z, the vertical lines denote +- λt.*

Cross-Validation for LASSO

The regularization parameter λ in LASSO governs the trade-off between model sparsity and fidelity to the training data. To systematically choose an optimal λ, a grid search is employed with k-fold cross-validation using mean squared error (MSE) as the evaluation metric.

Grid Construction: Firstly, a set of candidate λ must be defined. In order to systematically select a range of values that covers all the important candidates, the maximum number must be known. The approach proposed by Friedman et al. is followed [16], where the largest value of λ, denoted $\lambda_{\max}$, is the smallest value for which the entire LASSO solution is zero. It is given by:

$$\lambda_{\max} = \max_j \left| \mathbf{a}_j^T \mathbf{u} \right|, \tag{2.10}$$

where $\mathbf{a}_j$ is the j-th column of the design matrix $\mathbf{A}$, and $\mathbf{u}$ is the output vector.

A logarithmically spaced grid of J values is constructed between $\lambda_{\max}$ and a smaller value $\lambda_{\min} = \epsilon \lambda_{\max}$, where a typical choice is $\epsilon = 10^{-4}$:

$$\lambda_j \in [\lambda_{\max}, \epsilon \lambda_{\max}], \quad \text{log-spaced for } j = 1, \ldots, J.$$

This strategy ensures that a wide range of estimates, from nearly Least-Squares solutions with low regularization, up to significantly sparse ones, will be compared.

Cross-Validation with Mean Squared Error: For each λ_j, k-fold cross-validation is applied to estimate predictive accuracy. In this process:

- The dataset is randomly split into k folds, i.e. equal subsets.
- For each fold, the model is trained on $k-1$ folds and validated on the held-out fold.
- The mean squared error (MSE) on the validation data is recorded.

The MSE for fold i is computed as:

$$\text{MSE}^{(i)} = \frac{1}{N_i} \sum_{j=1}^{N_i} (u_j - \hat{u}_j)^2,$$

where $\hat{u}_j$ is the prediction for sample j in the i-th fold. The average MSE across all folds gives the performance metric for that λ.

The preference of MSE as an evaluation metric over Mean Absolute Error (MAE), is due to MSE being more sensitive to outliers. The use of MAE would result in variance underestimation.

The optimal regularization parameter is selected as the one minimizing cross-validated MSE:

$$\lambda^* = \arg\min_{\lambda_k} \text{CV-MSE}(\lambda_k).$$

LASSO Procedure: With a selected range of regularization parameter values, a k-fold Cross validation stage with multiple LASSO optimizations performed with ISTA, the optimal λ is found. With this value as input, a final LASSO is solved (again with ISTA) using the full dataset for regression. This constitutes the full LASSO procedure, and since some other LASSO variants will be explored, it will be referred to as *standard LASSO*.

2.3.5 Numerical Example

To illustrate the ability of the OMP and LASSO algorithms in finding the sparse solution to an undersampled rPCE problem, a simple numerical example is defined. A PCE case with 10D input is defined, including 2 important variables, while the remaining 8 contribute only small-amplitude noise. Firstly, the 2D function with the important variables is defined:

$$u(\boldsymbol{\xi}) = 5\,\xi_1 + 10\,\xi_2^2,$$

where $\xi_1, \xi_2 \sim \mathcal{N}(0,1)$.

In this simple example the PCE coefficients can be computed algebraically. The univariate orthonormal Hermite basis functions are

$$\psi_0(x) = 1, \qquad \psi_1(x) = x, \qquad \psi_2(x) = \frac{x^2 - 1}{\sqrt{2}},$$

and the multivariate basis is constructed as

$$\Psi_{(\alpha_1,\alpha_2)}(\xi_1, \xi_2) = \psi_{\alpha_1}(\xi_1)\,\psi_{\alpha_2}(\xi_2).$$

The function can be expressed directly in terms of the multivariate basis:

$$u(\xi_1, \xi_2) = 5\,\Psi_{(1,0)}(\xi_1, \xi_2) + 10\sqrt{2}\,\Psi_{(0,2)}(\xi_1, \xi_2) + 10\,\Psi_{(0,0)}(\xi_1, \xi_2).$$

From this expansion, the nonzero PCE coefficients are read off immediately:

$$c_{(0,0)} = 10, \quad c_{(1,0)} = 5, \quad c_{(0,2)} = 10\sqrt{2},$$

with all other coefficients equal to zero.

To make an example case with a sparse solution, 8 other Gaussian variables are

introduced, each of which affects the outcome via random amplitudes:

$$u(\xi) = 5\,\xi_1 + 10\,\xi_2^2 + \sum_{j=3}^{10} w_j\,\xi_j,$$

with each w_j drawn from $\mathrm{N}(0, 0.1)$, so it is different from evaluation to evaluation, to simulate an arbitrary, small-amplitude, noisy influence of the 8 variables, which should not show in the sparse solution.

With chaos order set as $p = 2$ the truncated basis in 10D contains

$$M = \binom{10+2}{2} = 66$$

orthonormal Hermite basis functions, with an equal number of unknown PCE coefficients. Two regression experiments were conducted:

- LASSO and OMP was performed using $N = [20, 30, 40, 50]$ LH samples.
- Least-squares Regression (OLS) was applied using $N = 66$ LH samples, where the square linear system is probably not adequate for coefficient estimation, and the noisy variables are likely to influence the otherwise simple model.

Firstly, the OLS rPCE solution is examined. As expected, the OLS is not immune to noise, especially considering that the system has as many equations as unknowns, since no oversampling is used. As illustrated in Fig. 2.6, coefficients that should be zero are activated and the important terms do not acquire the values of the analytical solution.

As far as the sparse PCE results are concerned, the only expansion that does not approximate the analytical solution is the OMP solution with 20 LH samples. OMP solutions using 30,40 and 50 samples, as well as all the LASSO expansions obtain zero or near-zero values in the terms that must be inactive and nearly identical values to the analytical solution in the important terms, demonstrating the sparse regression techniques' ability in finding the sparse solution.

In the subject of feature selection, it is important to note that while LASSO solutions obtain nonzero values on some terms that ideally would be zero, these are orders of magnitude smaller than the coefficients of interest, such that they can be considered negligible in the PCE solution. Therefore, the LASSO and cross-validation procedure successfully detected the important basis terms. The same can be said for OMP, excluding the first solution with 20 samples, whose failure indicates how OMP's greedy nature may render it less robust in severely undersampled applications (something that will also be illustrated in the case study of Chapter 3). The remaining OMP solutions activated the important terms and excluded the insignificant ones almost as well as the LASSO solutions. Fig. 2.7 illustrates the mean value of the coefficients that should be zero, in different sample sizes, where

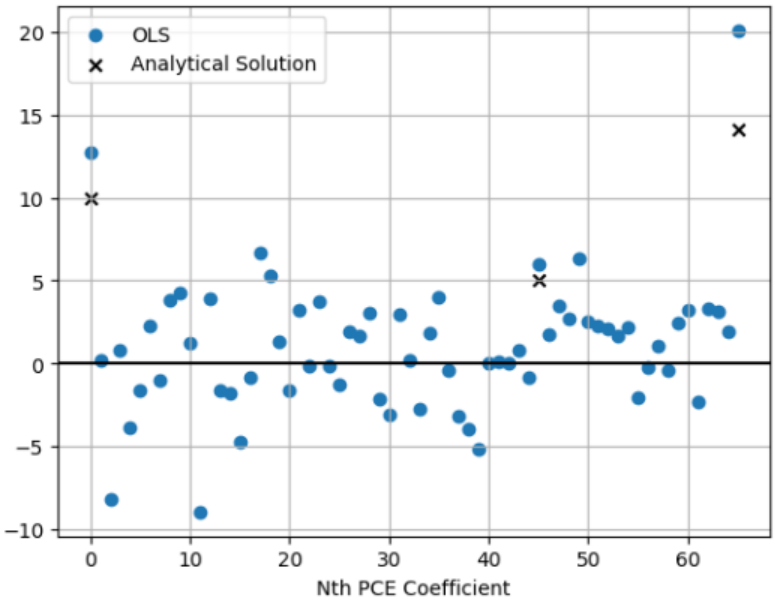

Figure 2.5: *Numerical Example: Comparison of OLS PCE Coefficients (blue) with the Analytical Solution (black).*

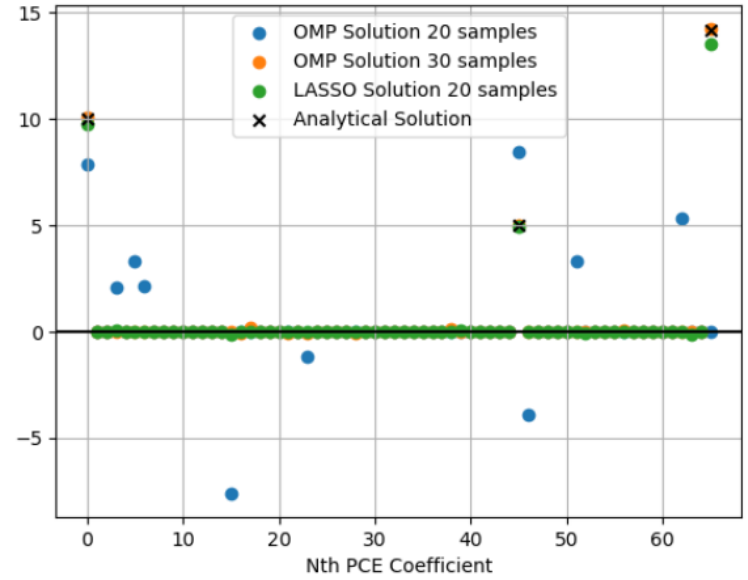

Figure 2.6: *Numerical Example: Comparison of OMP solution Coefficients with 20 samples (blue), OMP with 30 samples (orange), LASSO with 20 samples (green), with Analytical Solution (black).*

all sparse results perform well except for the first OMP expansion.

Of interest are the termination conditions for both OMP and LASSO solutions. All the ISTA executions for the LASSO method were terminated due to the solution being unchanged (ISTA ϵ set to $1e-6$) and never reached the maximum number of iterations, implying that the LASSO optimization problem is solved. For the

termination hyperparameters of OMP, the maximum sparsity level k was fixed to 10 terms (15% of the basis size) and the tolerance to $1e-6$. Even in the successful OMP solutions, the error target proved too low for the algorithm to stop in the activation of three terms, and small non-zero values were assigned to terms that should be zero, always yielding solutions with an L_0 norm of 10. However, as was in the LASSO case, these values are considered negligible and OMP solutions with sample size greater than 20 are good-quality approximations of the analytical solution. For this reason, cross-validation for the tuning the sparsity level and the error tolerance, while possible, was deemed unnecessary for OMP and was omitted in this study, giving this algorithm a large computational advantage over LASSO, where cross validation is necessary.

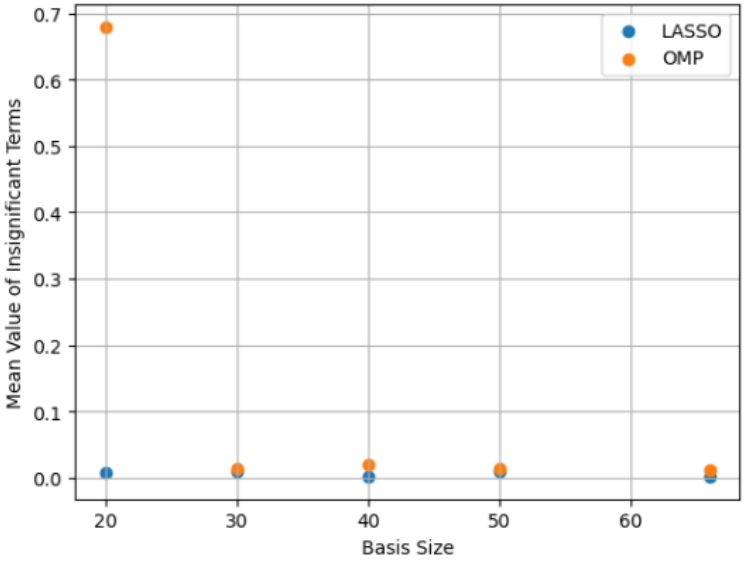

Figure 2.7: *Numerical Example: Mean Absolute Value of coefficients that should be zero for OMP (orange) and LASSO(blue) solutions of different sample sizes.*

Given that LASSO solutions across all sample sizes and OMP solutions for sizes greater than or equal to 30 select the correct features, it is important to evaluate how the accuracy of these features' coefficients changes with sample size. In Fig. 2.8 it is clear that, for both OMP and LASSO, results ameliorate and converge as sample size increases. For sample sets larger than 20 (where both approaches succeed), no method between LASSO and OMP is clearly outperforming the other.

In summary, the sparse algorithms were able to compute the sparse solution provided with the undersampled PCE experimental sets, with LASSO needing slightly fewer samples to produce an approximation of the analytical solution. Variables that should not be present in the sparse solution were excluded in the sparse expansions, while the OLS solution was inaccurate. In addition to this simple demonstration, this example served as a benchmark to test the LASSO software. The LASSO results, produced by the developed LASSO-ISTA C software were compared to those of the popular Python Library Scikit-learn [26], observing nearly identical output.

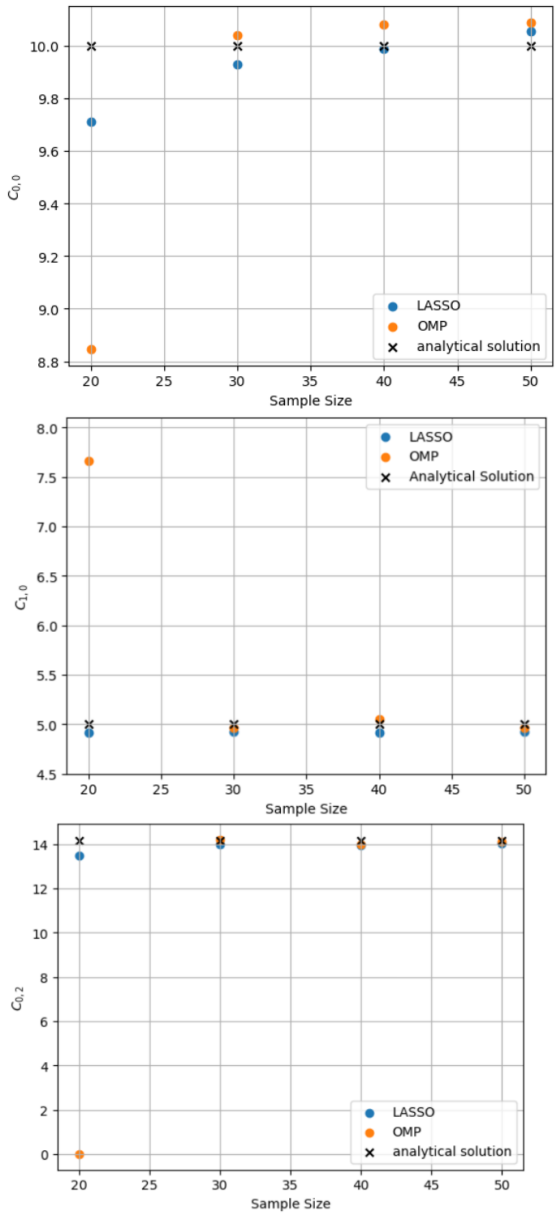

Figure 2.8: *Numerical Example: Predicted Coefficients by LASSO (blue), OMP (orange) and the analytical solution (black). The mean value (top), the coefficient of ξ_1 (middle) and the coefficient of ξ_2^2 (bottom) are presented.*

2.3.6 Conditions for Accurate Sparse Regression - the need for Basis reduction

As the previous numerical example shows, the accurate solution of the undetermined linear system is possible if the true solution is sparse. However, it is not guaranteed and an important part of CS theory has been coming up with conditions that determine how probable this recovery is. These depend on the design matrix, which, in turn, depends on the samples. The most fundamental property for the design

matrix is the Restricted Isometry Property [1]. Related to RIP is the coherence of the regression matrix [15], which is significantly high in cases of undersampled PCE, presenting a potential lack of robust results that needs to be tackled.

Restricted Isometry Property (RIP)

The Restricted Isometry Property (RIP) is a fundamental concept in CS [8] that ensures the design matrix $\mathbf{A}$ preserves the distance between sparse signals. A matrix $\mathbf{A}$ satisfies the RIP of order k if there exists a constant $\delta_k \in (0, 1)$ such that for all k-sparse vectors $\mathbf{x}$, the following inequality holds:

$$(1 - \delta_k)\|\mathbf{x}\|_2^2 \leq \|\mathbf{A}\mathbf{x}\|_2^2 \leq (1 + \delta_k)\|\mathbf{x}\|_2^2.$$

The RIP ensures that the measurement matrix $\mathbf{A}$ acts as a near-isometry on the set of sparse signals, which is crucial for accurate signal recovery, either by greedy or L_1 methods. It is however computationally infeasible to compute the δ_k value because it requires the use of all possible k-sparse vectors. For this reason, the coherence of the regression matrix is considered, which is related to RIP.

Coherence

Mutual coherence is another measure of the quality of the measurement matrix $\mathbf{A}$. The coherence μ of a matrix $\mathbf{A}$ is defined as:

$$\mu(\mathbf{A}) = \max_{1 \leq i,j \leq N,\, i \neq j} \frac{|\langle \mathbf{a}_i, \mathbf{a}_j \rangle|}{\|\mathbf{a}_i\|_2 \|\mathbf{a}_j\|_2},$$

where $\mathbf{a}_i$ and $\mathbf{a}_j$ are columns of $\mathbf{A}$. Coherence measures the largest correlation (inner product) between two normalized columns of $\mathbf{A}$. A lower coherence implies that the columns are more orthogonal or mutually incoherent, which is desirable for accurate sparse estimation. Contrary to the RIP constant, coherence is computationally simple to evaluate, and low coherence is a practical indicator of RIP-like behavior [15].

Specifically, it has been shown ([30]) that for an $m \times n$ matrix $\mathbf{A}$, Restrictive Isometry Principle for k sparse solutions is guaranteed if:

$$k < \frac{1}{2}\left(1 + \frac{1}{\mu(\mathbf{A})}\right). \tag{2.11}$$

Although this is a sufficient (but not necessary) condition and may be overly strict or difficult to satisfy in high-dimensional PCE settings, it nonetheless highlights the practical importance of designing low-coherence measurement matrices. Keeping coherence low increases the likelihood of accurate sparse estimation, even when RIP cannot be verified directly.

In the numerical example of 2.3.5, the coherence of the different design matrices is examined, and it is obvious that the limit of eq. (2.11) is stricter than needed. The solution of the numerical example contained three terms, meaning that coherence should be as low as 0.2. As it can be seen in Fig. 2.9, coherence values exceed that limit, but as it was described, the sparse solutions were good quality approximations of the analytical solution and both OMP and LASSO were successful. Nevertheless, it was also evident that with the increase in sample size results ameliorated, and as illustrated Fig. 2.9, the increase of sample size lowers the coherence of the regression matrix. Furthermore, the design matrix coming from 20 LH samples exhibits significantly high coherence, serving as an indication to OMP's failure in that case.

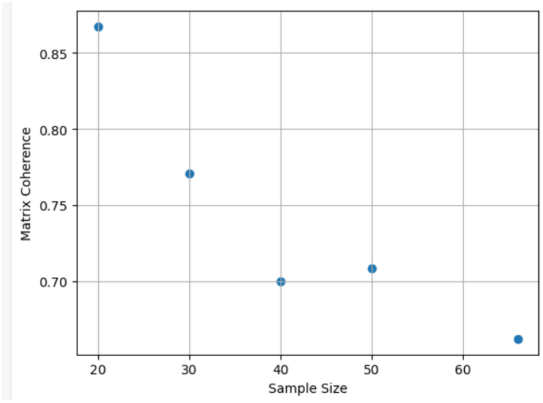

Figure 2.9: *Numerical Example: Mutual coherence $\mu(A)$ vs. sample size.*

The problem with PCE

The foundations of CS are rooted in signal-processing, where the full (square) linear regression system (e.g of Nyquist samples) can be solved exactly and serves as the gold-standard benchmark for sparsity-promoting methods that use fewer samples. In contrast, in rPCE, as already mentioned, oversampling is critical for the accuracy of the solution.

One key consequence of this difference is in the mutual coherence of the design matrix. In a square signal-processing regression problem, the design matrix is orthonormal and hence has zero coherence. However, neither square nor undersampled PCE design matrices enjoy this property: their columns are more strongly correlated, which degrades sparse-recovery guarantees. For example, this differnce in coherence can be shown when comparing randomly undersampled matrices from Discrete Cosine Transform (DCT) applications and Hermite matrices from undersampled LHS sets. DCT is a common basis for signal-processing and CS applications and therefore is an appropriate case to compare with PCE. In particular, when the system is undersampled, the coherence of Hermite-PCE matrices falls off much more slowly with the number of samples than does the coherence of a randomly sub-sampled DCT basis.

Figure 2.10 plots the mutual coherence $\mu(A)$ versus the sample size for three cases:

- Randomly sub-sampled DCT in $P = 45$ Nyquist size (blue),
- Hermite-PCE in $d = 8, p = 2, P = 45$ using Latin Hypercube samples of the joint Gaussian distribution (orange), and
- Hermite-PCE in $d = 8, p = 3, P = 165$ using Latin Hypercube samples of the joint Gaussian distribution (green).

At every sample size, the DCT design achieves substantially lower coherence than either Hermite case.

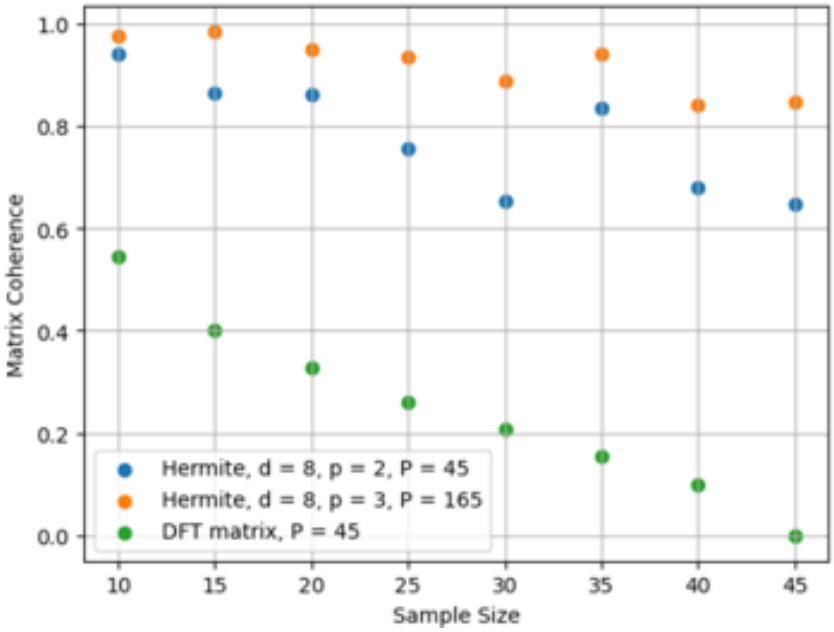

Figure 2.10: *Mutual coherence $\mu(A)$ vs. sample size for DCT (green) and Hermite-PCE in 8D with $p = 2$(blue) and 8D with $p = 3$ (orange).*

Basis Reduction Strategies

It is also clear from Figure 2.10 that the quality of design matrices for undersampled PCE decreases with the rise in basis size. This is expected since the large number of matrix columns increases the probability of correlated pairs. Thus, reducing the basis size can substantially lower coherence and improve stability of the sparse solution. This motivated the development of reduced-basis sparse methods, especially for PCE, where coherence is high. The reduction of basis terms can be done a-priori or produced via the sparse regression algorithm, during the solution.

1. A-priori Basis Reduction

Hyperbolic truncation schemes [7] discard high order terms and terms with variable interactions. For a user-defined variable q, the reduced basis set is defined as:

$$\mathcal{I}_{p,q} \;=\; \Big\{\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_d) \in \mathbb{N}_0^d : \|\boldsymbol{\alpha}\|_q \leq p\Big\} \;=\; \Big\{\boldsymbol{\alpha} : \sum_{i=1}^{d} \alpha_i^q \leq p^q\Big\} \tag{2.12}$$

To avoid another hyperparameter (q-norm) and to demonstrate the performance of sparse PCE compared to OLS-based PCE, hyperbolic truncation was not used in this study.

2. Basis reduction in the regression algorithm

The LASSO method does not introduce any basis reduction and considers the whole set of the matrix columns before suppressing those that its solution dictates. On the other hand, OMP relies on reducing the basis size greedily, which may result in the omission of important terms and decrease robustness. It would be preferable, if the selection of a basis subset would result from the consideration of the whole set and then a design matrix of lower coherence could be produced.

Two stage LASSO algorithms

Basis reduction can be achieved with the help of L_1 minimization and the absence of greedy-related pitfalls. Certain methods exist that use a LASSO regression as a first step, in order to determine the basis subset containing the important terms. Since the whole polynomial basis is considered in a LASSO solution, the selection of an important subset may be impartial.

Firstly, the LASSO problem is solved on the full PCE basis to obtain an initial active set S_1, i.e the basis functions with non-zero coefficients. The polynomial basis is then restricted to only the terms in S_1 and the regression problem is re-solved. The basis is drastically cut down, due to the sparsity in the LASSO solution, and the mutual coherence of the resulting submatrix is much lower, potentially yielding more stable recovery and reduced estimation error.

After obtaining the active set S_1 from the first-stage LASSO, the solution can be acquired with one of two ways:

1. **Post-LASSO OLS (pLASSO):** [5] If the solution of the first LASSO is significantly sparse, the new problem may be overdetermined enough that OLS on the reduced support can be solved

   $$c_{\text{OLS}} = \arg\min_{cS_1} \; \|u - A_{S_1}\, c_{S_1}\|_2^2.$$

   The problems with sparse regression do not affect the final coefficient values, and an oversampled OLS (with the same sample set as before) can produce a better result.

2. **Relaxed LASSO (rLASSO):** [22] A second LASSO can be solved on S_1, whose design matrix has likely much smaller coherence, using the λ penalty

chosen in stage 1 by cross-validation as maximum for the cross-validation of the second step (λmax):

$$c_{\text{relax}} = \arg\min_{c_{S_1}} \frac{1}{2}\|u - A_{S_1}\, c_{S_1}\|_2^2 + \lambda \sum_{j \in S_1} |c_j|, \quad \lambda \in \left[10^{-4}\,\lambda_{\max},\ \lambda_{\max}\right],$$

With the reduced basis and lower coherence, LASSO now is more likely to accurately estimate the sparse solution.

It is important to note that the standard LASSO method is of significantly higher computational cost than OMP, due to Cross-Validation. Therefore, the two-stage LASSO approaches require even more computational cost. However, this augmentation of computational cost may be negligible if the model of the UQ case is intensive, the two-stage approaches are yield more accurate results, and sample economy is of much greater value than the computational efficiency of the sparse algorithm. In other words, if a more tedious approach in the regression stage allows for further reduction of sample size, it is advantageous. For the sake of clarity, it is noted that sampling ratios (SR) for the reduced basis approaches refer to the initial truncated basis size, for a fair comparison.

The performance of the standard LASSO, OMP as well as of the two-stage LASSO methods, will be analyzed in the first case, and compared for the aerodynamics-related cases (Cases 2 and 3).

Chapter 3

Applications

3.1 Case 1 - Borehole Function

In order to evaluate the ability of sparse rPCE methods on undersampled settings to compute accurate UQ results, a test function with both a considerable number of uncertain variables and nonlinearity in the QoI seems an appropriate choice before moving on to applications in aerodynamics.

3.1.1 Problem Formulation

The Borehole function [17] models water flow through a borehole drilled through two aquifers, given by the analytical expression:

$$Q = \frac{2\pi T_u (H_u - H_l)}{\ln\left(\frac{r}{r_w}\right)\left[1 + \frac{2LT_u}{\ln\left(\frac{r}{r_w}\right) r_w^2 K_w} + \frac{T_u}{T_l}\right]} \tag{3.1}$$

To define a UQ problem, a distribution of uniform type was defined for each one of 8 stochastic variables:

- $r_w \in \mathcal{U}[0.05, 0.15]$ m: Borehole radius
- $r \in \mathcal{U}[100, 50000]$ m: Radius of influence
- $T_u \in \mathcal{U}[63070, 115600]$ m^2/yr: Upper aquifer transmissivity
- $H_u \in \mathcal{U}[990, 1110]$ m: Upper aquifer potentiometric head
- $T_l \in \mathcal{U}[63.1, 116]$ m^2/yr: Lower aquifer transmissivity
- $H_l \in \mathcal{U}[700, 820]$ m: Lower aquifer potentiometric head
- $L \in \mathcal{U}[1120, 1680]$ m: Borehole length

- $K_w \in \mathcal{U}[9855, 12045]$ m/yr: Borehole hydraulic conductivity

The Chaos Order for the truncation of basis functions was set to $p = 2$. This value is commonly used in rPCE for engineering problems. According to the total-order truncation formula (eq. (2.4)) for $d = 8$ dimensions and a chaos order of $p = 2$ the number of PCE terms is: $P = 45$.

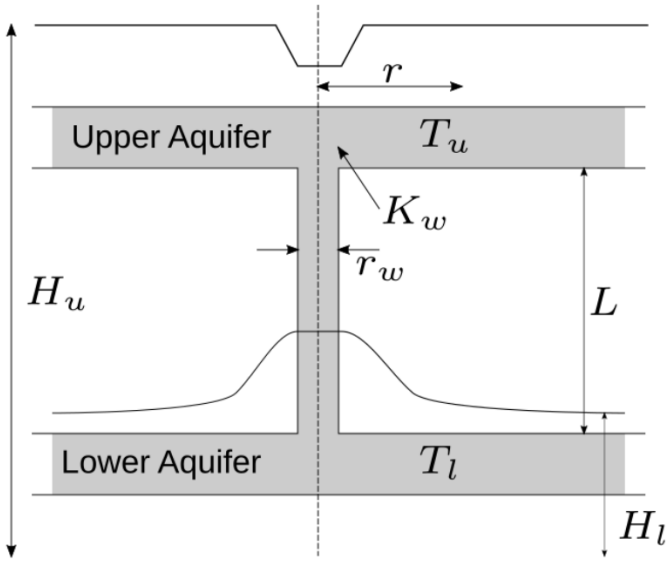

Figure 3.1: *Case 1: Borehole flow illustration, adapted from [17].*

Since the aerodynamic cases to be examined are of Gaussian type and software for this type of application was developed, the uncertain variables are first mapped to independent standard Gaussian random variables. Specifically, the Borehole parameters are transformed into a set of eight variables $\xi_1, \ldots, \xi_8 \sim \mathcal{N}(0, 1)$, where $\mathcal{N}(0, 1)$ denotes the standard normal distribution with mean zero and unit variance. These ξ_i serve as the stochastic inputs to the PCE. Their corresponding physical parameters (denoted by θ_i below) are then recovered by applying the transform:

$$\theta_i = a_i + (b_i - a_i)\,\Phi(\xi_i), \tag{3.2}$$

where $\Phi(\cdot)$ is the cumulative distribution function (CDF) of the standard normal distribution, namely:

$$\Phi(\xi) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\xi} e^{-t^2/2}\, dt. \tag{3.3}$$

3.1.2 MC Simulation

Due to the low computational cost of the Borehole function, a MC simulation is non-prohibitive. Statistical moments were acquired using $N = 500000$ samples, yielding the values of Table 3.1.

Table 3.1: *Case 1: MC Simulation Results*

Statistic	Value
Mean	134.39724
SD	54.6927

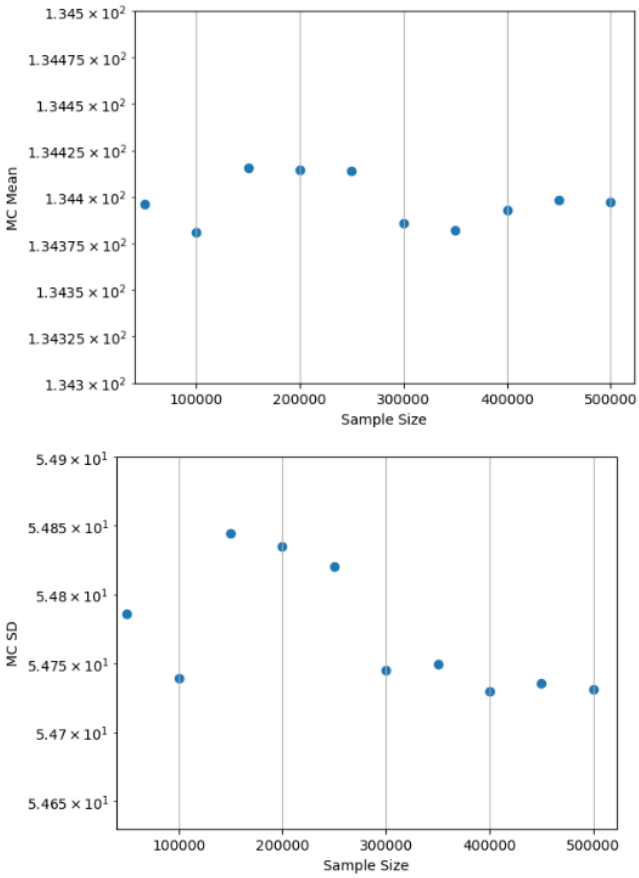

Figure 3.2: *Case 1: Results in terms of the number of samples (N) processed by MC. Mean (Top) and SD (bottom).*

The evaluation of PCE solutions is now possible, since the mean value and SD of any PCE solution (sparse or non-sparse) can be compared with the values of Table 3.1, considered to be the "ground truth". While the mean is generally easy to approximate and the SD is a better indicator of the quality of a UQ method, accurate moment estimation alone may be misleading. There is always a possibility that an expansion may approximate the statistical moments well by activating, however, the wrong basis terms (eq. (2.5)), which might lead to wrong approximations at points of interest, generating thus a non-dependable surrogate for the analytical expression (eq. (3.1)). To avoid this, predictive accuracy is also evaluated through the Mean

Absolute Relative Error (MARE) on a randomly selected test set:

$$\text{MARE} = \frac{1}{N_{\text{test}}} \sum_{i=1}^{N_{\text{test}}} \frac{|u_i - \hat{u}_i|}{|u_i|},$$

where u_i and $\hat{u}_i$ are the "ground truth" and predicted QoIs (by any of the PCE models)of the selected test samples. Thus, both reliable moment estimation and low MARE are required for a good PCE model.

With the reference values known from the MC simulation, an OLS PCE solution is computed. By gradually oversampling the case with LHS sets and repeating the same computation of the PCE expansion, the statistical moments and the MARE (for a randomly selected test set of 45 samples) have been computed. The same test set will also be used in the evaluation of the sparse expansions. With the OLS PCE method, runs showed that acceptable results can be obtained with as many as $N = 125$ function calls, resulting to overampling with $SR = 2.7$ (Table 3.2).

Table 3.2: *Case 1: OLS ($N = 125$) PCE Metrics*

Statistic	**MC**	**OLS ($N = 125$)**
Mean	134.39724	134.605411
SD	54.6927	55.41519
MARE	—	0.077

Examining the OLS PCE solution (Fig. 3.3) there are important differences in coefficient magnitude. If only 8 of the 45 PCE coefficients (those with the larger magnitude) are kept by repeating the OLS PCE run and retaining only the corresponding bases, the SD changes from 55.41519 to 53.71927, which is considered to be a comparatively small change. This is an encouraging outcome and implies that sparse regression on undersampled sets is expected to reduce the computational cost (much less function calls) without practically damaging the computed statistical moments. One should keep in mind that, in the above, the user has firstly to run a full regression PCE to select the most important bases, and then run least squares for the selected bases, but this does not make any sense since it increases, rather than decreases, the computational cost, so we definitely need sparse PCE methods that can simultaneously select the bases and compute the reduced PCE model using a reduced number of samples (undersampling). The ranking of the most important terms' coefficients (Table 3.3) shows that the borehole diameter (r_w) and the potentiometric head of the upper aquifier (H_u) are the most influential, with their univariate terms as well as their interaction term being present. The two least influential uncertain variables are the radius of influence (r) and borehole length (L).

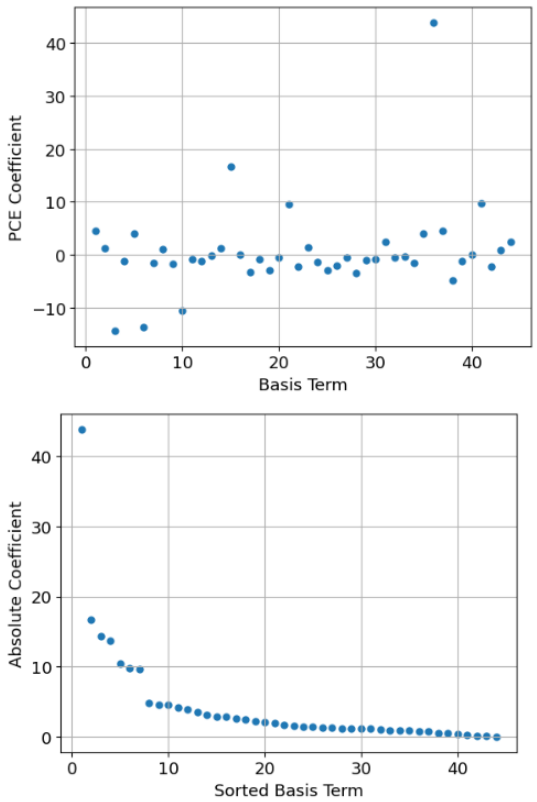

Figure 3.3: *Case 1: The computed 44 OLS PCE coefficients (the 0th one is omitted) using $N = 125$ function calls (top). The same coefficients (absolute value) are re-plotted in descending order (bottom). It is clear (plot at bottom) that there are 7 PCE coefficients which are the most important compared to the rest.*

Table 3.3: *Dominant PCE terms for Case 1 (Borehole function).*

PCE Coefficient	Multi-index	Gaussian Variable	Borehole Variable
134.60541	(0,0,0,0,0,0,0,0)	Constant	Constant
43.82374	(1,0,0,0,0,0,0,0)	ξ_1	r_w
16.67048	(0,0,0,1,0,0,0,0)	ξ_4	H_u
-14.28694	(0,0,0,0,0,0,1,0)	ξ_7	L
-13.63271	(0,0,0,0,1,0,0,0)	ξ_5	T_l
-10.46278	(0,0,0,0,0,1,0,0)	ξ_6	H_l
9.76451	(1,0,0,1,0,0,0,0)	$\xi_1\xi_4$	$r_w H_u$
9.62043	(0,0,1,0,0,0,0,0)	ξ_3	T_u

3.1.3 Use of Sparse Methods

Firstly, the different sparse regression methods are used to compute PCE solutions, by processing three LHS sets of sizes $N = 20, 30, 40$, corresponding to $SR = 0.44, 0.67, 0.89$ respectively. As one may see, all three of them correspond to undersampling conditions ($N < 45$). The statistical moments computed from sparse

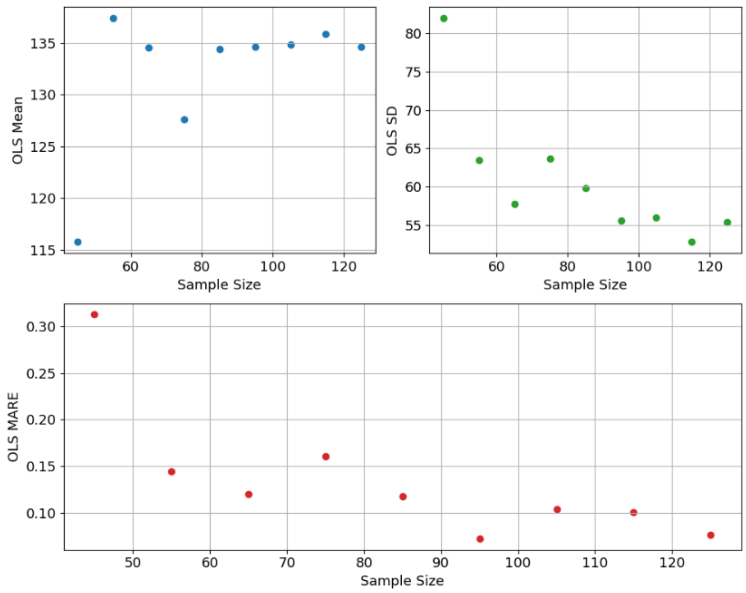

Figure 3.4: *Case 1: Computed Values of the OLS PCE Mean (top-left), SD (top-right), and MARE (bottom), using the OLS method, for various numbers of N samples (function calls).*

expansions based on $N = 20, 30, 40$ samples are shown in Tables 3.4 and 3.5.

Table 3.4: *Case 1: Sparse PCE predictions of Mean value* $N = 20, 30, 40$ *function calls.*

Method	$N = 20$	$N = 30$	$N = 40$
OMP	132.04120	137.66399	133.77910
LASSO	129.79700	134.26191	129.73971
rLASSO	129.83148	134.60418	129.69034
pLASSO	129.84134	135.02873	129.46614
MC Mean: 134.39724			

Table 3.5: *Case 1: Sparse PCE predictions of SD* $N = 20, 30, 40$ *function calls.*

Method	$N = 20$	$N = 30$	$N = 40$
OMP	61.04043	56.99045	57.48194
LASSO	52.70903	47.82519	56.71651
rLASSO	53.46150	52.55117	56.83270
pLASSO	53.69978	53.15840	57.43193
MC SD: 54.6927			

Sparse PCE Moments

The Mean and SD predictions for all the sample sets are illustrated in Fig. 3.5 and 3.6. Generally, it is expected (but not guaranteed) that the higher the sample size, the better the "ground truth" values are approximated. OMP seems to do a better job in predicting the mean value of the QoI, while it is not clear whether a method is consistently better in SD prediction in this instance. Statistical moments predicted by rLASSO and pLASSO are significantly similar with each other and their values do not fluctuate as much as those of OMP and LASSO, having however a significant deviation from the MC value.

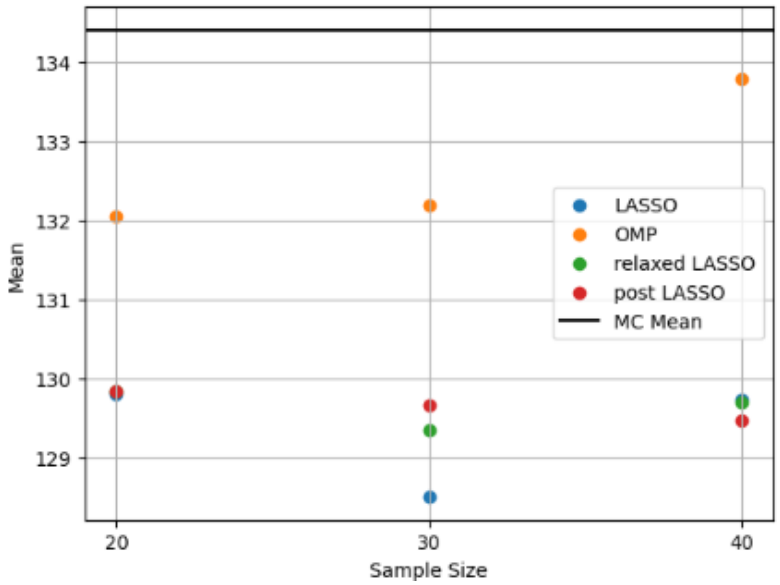

Figure 3.5: *Case 1: Sparse PCE predictions of the mean value of the QoI for* $N = 20, 30, 40$ *samples. Results for LASSO (blue), OMP (orange), rLASSO (green) and pLASSO (red) are compared with the MC mean (black line).*

Test Set Prediction

As far as the prediction of the random test set is concerned, OMP shows the worst performance in MARE values (Fig. 3.7). The difference of the methods' predictive capabilities can also be seen in Fig. 3.8 where the predicted QoI values from the sparse PCE are compared with the sorted values of the test set. OMP is the only method out of the four that is purely based on OLS solutions and is not influenced by any form of Cross-Validation, which explains why its expansions showcase the worst predictive behavior, but also why it has the lowest computational cost. The computational advantage, however, of OMP is not important in cases where the evaluation cost is high.

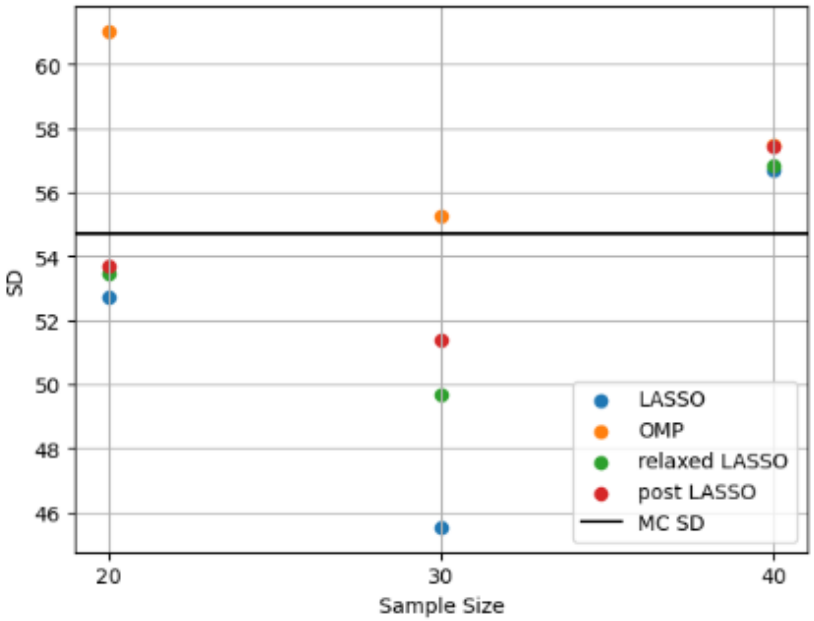

Figure 3.6: *Case 1: Sparse PCE predictions of SD for $N = 20, 30, 40$ samples. Results for LASSO (blue), OMP (orange), rLASSO (green) and pLASSO (red) are compared with the MC SD (black line).*

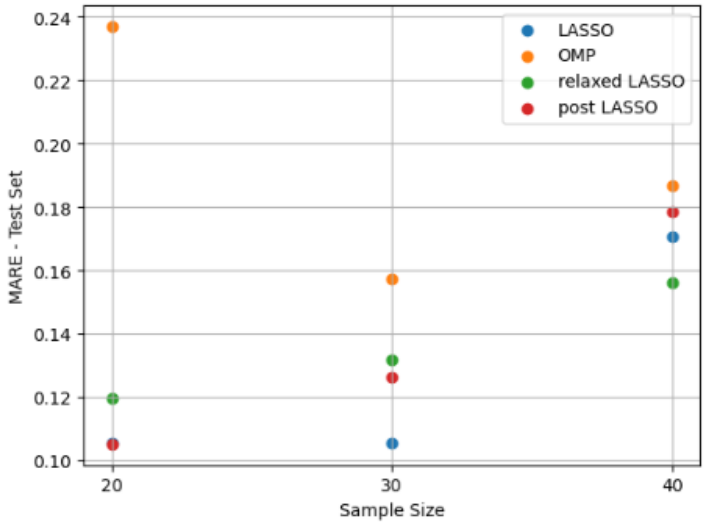

Figure 3.7: *Case 1: MARE values from the prediction of the random test set. Results are shown for LASSO (blue), OMP (orange), rLASSO (green) and pLASSO (red) expansions.*

Sparse PCE Coefficients

The accuracy of the sparse PCE coefficients is examined. Firstly, the comparison of the expansions with the OLS ($N = 125$) expansion is illustrated in Fig 3.9. It is evident that unimportant terms are rarely activated. OMP seems to be the method that is most likely to activate and overestimate unimportant features.

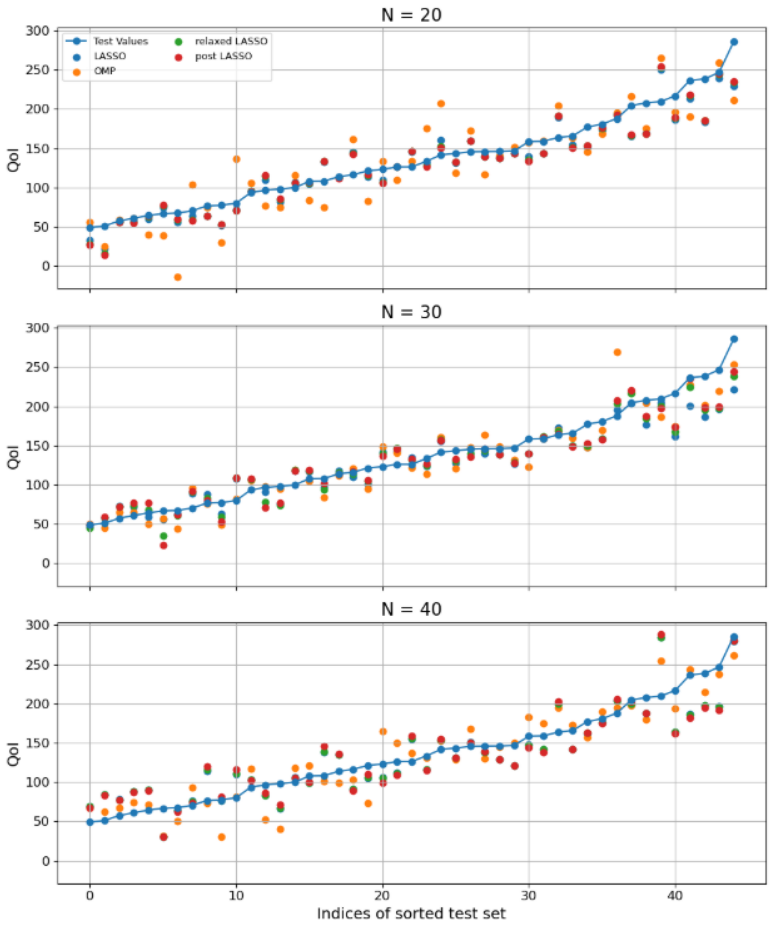

Figure 3.8: *Case 1: Predicted QoI values by expansions, computed from LH sets of size 20 (top), 30 (middle), and 40 (bottom) (single LH set shown). Predictions obtained from LASSO (orange), OMP (green), rLASSO (red) and pLASSO (purple) are compared to the sorted test QoI values (blue).*

As far as the dominant coefficients of 3.3 are concerned, the sparse PCE coefficients of the corresponding terms are compared with the OLS ($N = 125$) results in Tables 3.6, 3.7 3.8. First, the same features that LASSO activates are activated by rLASSO and pLASSO, which is expected since their reduced basis comes from LASSO. OMP misses important terms more frequently than the other two highlighting how its greedy nature may result in problems during basis selection. For all the sparse methods, the most difficult basis term to activate is the one with variable interaction ($r_w H_u$). Overall, the number of important coefficients that are kept by sparse PCE is greater when sample size (N) is greater and coefficients with larger magnitudes are more consistently approximated.

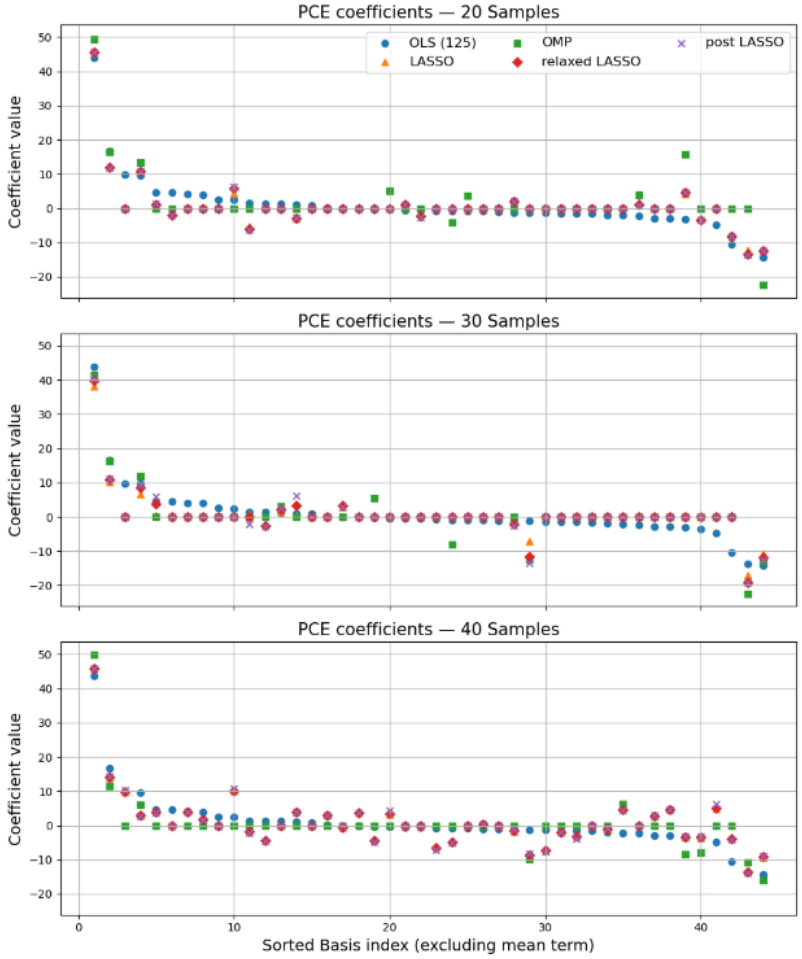

Figure 3.9: *Case 1: PCE coefficients from LH sets of size $N = 20$ (top), $N = 30$ (middle), and $N = 40$ (bottom) (single LH set shown). Solutions obtained from LASSO (orange), OMP (green), rLASSO (red) and pLASSO (purple) are compared to the OLS solution of $N = 125$ samples (blue). The basis terms are ordered so that the OLS expansion is sorted.*

Table 3.6: *Case 1: Dominant PCE coefficients for $N = 20$ samples.*

Variable	OLS (125)	LASSO	OMP	rLASSO	pLASSO
Constant	134.60541	129.79700	132.04120	129.83148	129.84134
$\xi_1(r_w)$	43.82374	45.29723	49.49111	45.53064	45.59736
$\xi_4(H_u)$	16.67048	12.43989	16.49772	11.90102	11.74702
$\xi_7(L)$	-14.28694	-12.02545	-22.47253	-12.38711	-12.49048
$\xi_5(T_l)$	-13.63271	-12.15403	0.00000	-13.27839	-13.59976
$\xi_6(H_l)$	-10.46278	-8.17673	0.00000	-8.15487	-8.14862
$\xi_1\xi_4(r_w H_u)$	9.76451	0.00000	0.00000	0.00000	0.00000
$\xi_3(T_u)$	9.62043	10.72800	13.49208	10.82414	10.85163

Table 3.7: *Case 1: Dominant PCE coefficients for $N = 30$ samples.*

Variable	OLS (125)	LASSO	OMP	rLASSO	pLASSO
Constant	134.60541	128.50583	132.18683	129.33844	129.66759
$\xi_1(r_w)$	43.82374	38.10129	41.39600	39.77577	40.21008
$\xi_4(H_u)$	16.67048	10.17371	16.34657	10.90124	11.08653
$\xi_7(L)$	-14.28694	-10.70003	-12.90204	-11.77124	-12.63863
$\xi_5(T_l)$	-13.63271	-17.03072	-22.46348	-19.10447	-19.20202
$\xi_6(H_l)$	-10.46278	0.00000	0.00000	0.00000	0.00000
$\xi_1\xi_4(r_w H_u)$	9.76451	0.00000	0.00000	0.00000	0.00000
$\xi_3(T_u)$	9.62043	6.65012	11.94291	8.56902	9.36017

Table 3.8: *Case 1: Dominant PCE coefficients for $N = 40$ samples.*

Variable	OLS (125)	LASSO	OMP	rLASSO	pLASSO
Constant	134.60541	129.73971	133.77910	129.69034	129.46614
$\xi_1(r_w)$	43.82374	45.89034	49.95365	45.87930	45.86002
$\xi_4(H_u)$	16.67048	13.93794	11.45381	14.03383	14.42260
$\xi_7(L)$	-14.28694	-9.24736	-16.02597	-9.16721	-8.85309
$\xi_5(T_l)$	-13.63271	-13.44677	-10.77520	-13.51446	-13.79977
$\xi_6(H_l)$	-10.46278	-3.73596	0.00000	-3.79763	-4.05277
$\xi_1\xi_4(r_w H_u)$	9.76451	9.81284	0.00000	9.91108	10.37662
$\xi_3(T_u)$	9.62043	3.19169	5.98320	3.08780	2.68464

3.1.4 Multiple LHS Sets Analysis

The previous analysis may provide some indicative results, but the negligible computational cost of Case 1 allows for a thorough comparison of the four sparse PCE approaches. The following procedure is applied. For each sample size $N \in \{15, 20, 25, 30, 35, 40, 45\}$:

(1) 50 different LH samples of size N are generated.

(2) A sparse PCE is fitted via LASSO, OMP, pLASSO and rLASSO on each LHS set of size N.

(3) For each N, the average moments and MARE are computed from the resulting expansions.

Results (Figures 3.10, 3.11, 3.12) show that there are clear differences in the strengths and weaknesses of each method. With OMP performing well, only on Mean predictions, where it is the most accurate method, LASSO having the best MARE performance but struggling in both SD and Mean, pLASSO and rLASSO seem the two best methods in having satisfiable results for all metrics. The fact that pLASSO gives higher SD predictions than rLASSO is due to it being produced by an OLS solution, whereas coefficients are shrunk in rLASSO. For the same reason OMP

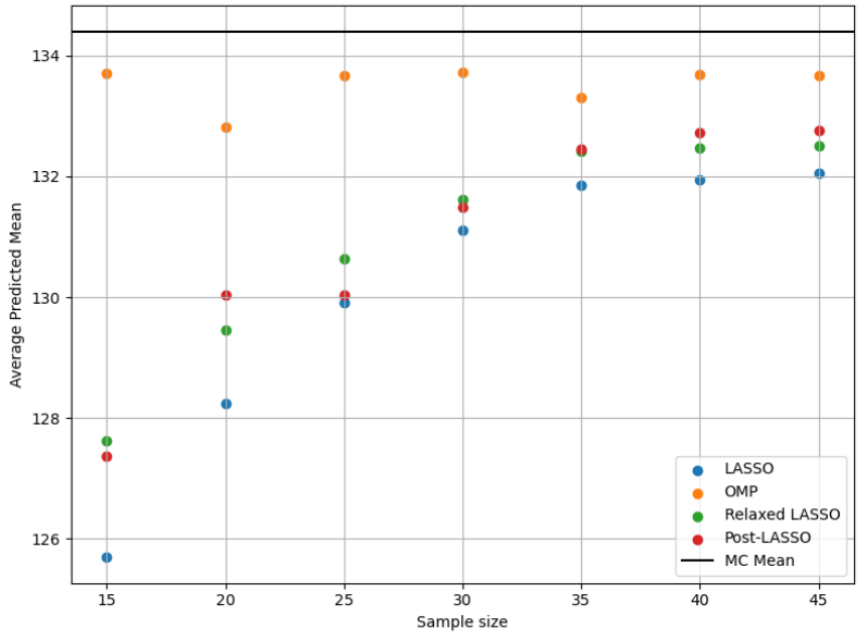

Figure 3.10: *Case 1: Average predicted Mean. Results from LASSO (blue), OMP (orange), rLASSO (green) and pLASSO (red) are compared with the MC Mean (black line).*

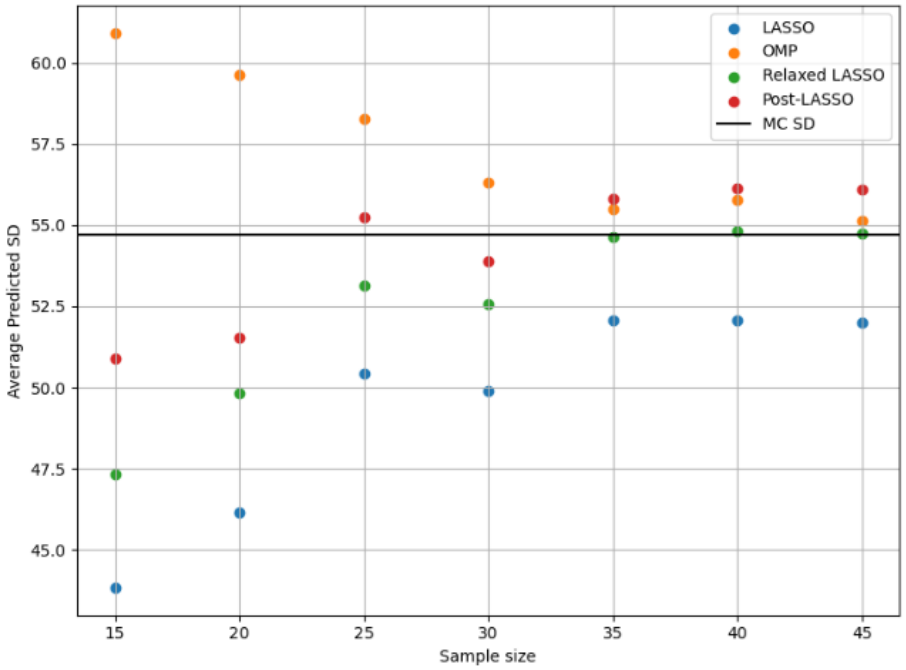

Figure 3.11: *Case 1: Average predicted SD. Results from LASSO (blue), OMP (orange), rLASSO (green) and pLASSO (red) are compared with the MC SD (black line).*

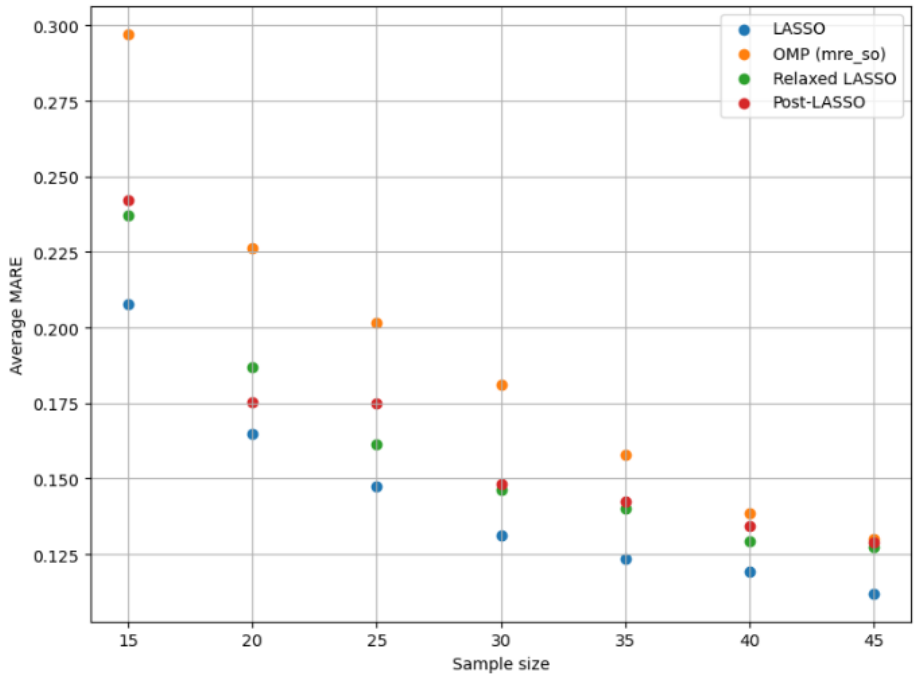

Figure 3.12: *Case 1: Average prediction MARE on the random test set. 50th (top) and 90th (bottom) Percentile. Results from LASSO (blue), OMP (orange), rLASSO (green) and pLASSO (red).*

significantly overestimates the SD, while standard LASSO underestimates the SD.

3.1.5 Conclusions from Case 1

As far as Case 1 is concerned, it has become evident that:

- The Borehole function, with $d = 8$ uncertain variables and significant non-linearity, constitutes a challenging regression problem for UQ (high MAREs, MC sample size, OLS oversampling etc.). Regardless of its straightforward formulation and implementation the complexity of the function is what disturbs the quality of PCE results.
- Due to the complexity of the function, the comparison of the sparse methods was not initially clear when 3 sets of difference sizes were used, and making the sample size (N) larger did not guarantee improvement in PCE performance.
- The strengths and weaknesses of each sparse method became clear once multiple sets were used for each size and average performance was considered.
- Sparsity-promoting algorithms can produce satisfactory results by using less samples than the unknowns (undersampling).
- OMP tends to struggle in predictive accuracy, while both OMP and standard LASSO show weaknesses in estimating the SD consistently.

- Standard LASSO also exhibits reduced accuracy in estimating the mean compared to other sparse approaches.
- Among the tested methods, pLASSO and rLASSO, while producing estimates with deviation from the MC values, seem to have the most balanced performance in computing the statistical moments and predicting the test set.

3.2 Case 2 - Aerodynamic Analysis with Flow Uncertainties

3.2.1 Problem Formulation

In this study, XFOIL is employed to perform an UQ analysis of a NACA 2412 airfoil, with a focus on the aerodynamic efficiency as the QoI. The model is subjected to flow uncertainties, modeled as independent Gaussian random variables:

- Freestream velocity $U \sim \mathcal{N}(25 \text{ m/s},\ 0.5^2)$
- Angle of attack $\alpha \sim \mathcal{N}(1.5^\circ,\ 0.1^2)$
- Dynamic viscosity $\mu \sim \mathcal{N}(1 \times 10^{-5},\ (0.1 \times 10^{-5})^2)$
- Air density $\rho \sim \mathcal{N}(1.5,\ 0.2^2)$

The QoI is the Lift-to-Drag ratio of the airfoil (C_L/C_D). Due to the low-dimensionality of the input variables, using $p = 3$ as Chaos Order is computationally feasible. While accurate results are likely captured with $p = 2$, the number of basis terms introduced by $p = 3$ make the basis large enough for sparse rPCE to be useful compared to OLS rPCE. The number of basis polynomials and unknown coefficients is computed $P = 35$ (eq. (2.4)).

The goal is to propagate these input uncertainties through XFOIL and analyze their effect on the C_L/C_D response using the sparse promoting techniques on the rPCE. It is examined whether compressibility or sparsity on the PCE solution can be taken advantage of for sample economy. The success of sparse PCE on this simple aerodynamic case motivates their use on a more complex case, with a model of higher fidelity (Case 3).

3.2.2 Simulation Parameters

The flow around an isolated NACA 2412 airfoil (Fig. 3.13) is simulated. Since XFOIL models incompressible flow, its input consists of angle of attack and Reynolds number. Hence, the velocity, density and viscosity components are firstly used to calculate the Reynolds number. While the Reynolds number could be treated as an uncertain variable, omitting velocity, density and viscosity, their treatment as uncertain input and the use of the Reynolds number as a dependent parameter is useful for two reasons: (a) the influence of these parameters due to their difference in orders of magnitude is of interest, (b) their use is crucial for the case to have a large enough number of unknowns for sparsity to be useful. The *Reynolds number*, Re, is a dimensionless quantity defined by

$$Re \ = \ \frac{\rho\, v\, L}{\mu} \ = \ \frac{v\, L}{\nu} \tag{3.4}$$

where:

ρ fluid density, $\mathrm{kg\,m^{-3}}$

v characteristic velocity, $\mathrm{m\,s^{-1}}$

L characteristic length scale, m

μ dynamic viscosity, $\mathrm{Pa\,s}$

$\nu = \mu/\rho$ kinematic viscosity, $\mathrm{m^2\,s^{-1}}$

The Reynolds number represents the ratio of inertial forces to viscous forces in a flow, and it is an indicator of whether the flow is laminar or turbulent. In this case, the baseline Reynolds number is 3700000, which lies in the turbulent region.

XFOIL implementation

XFOIL is a widely used, moderate-fidelity aerodynamic analysis tool designed for the prediction of airfoil performance. It solves the coupled inviscid/vortex panel method and boundary layer equations [12], which simulate incompressible flows, and can accurately model the flow around isolated airfoils in steady conditions of subsonic flight. Despite its simplicity compared to high-fidelity CFD solvers, XFOIL captures key flow features and is especially effective for the range of Reynolds numbers associated with small, subsonic aircraft, such as Unmanned Aerial Vehicles (UAVs).

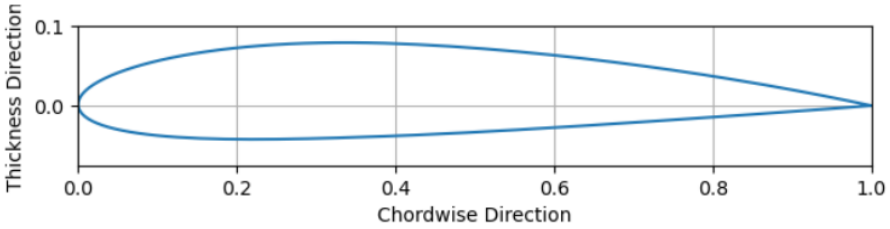

Figure 3.13: *Geometry of the NACA 2412 airfoil.*

3.2.3 MC simulation

The computational cost of an XFOIL simulation, while higher than that of Case 1 is still manageable for a MC simulation. Using $N = 30000$ samples, it yields the moments:

Table 3.9: *Case 2: MC Simulation Results*

Statistic	Value
Mean	80.46513
SD	2.39540

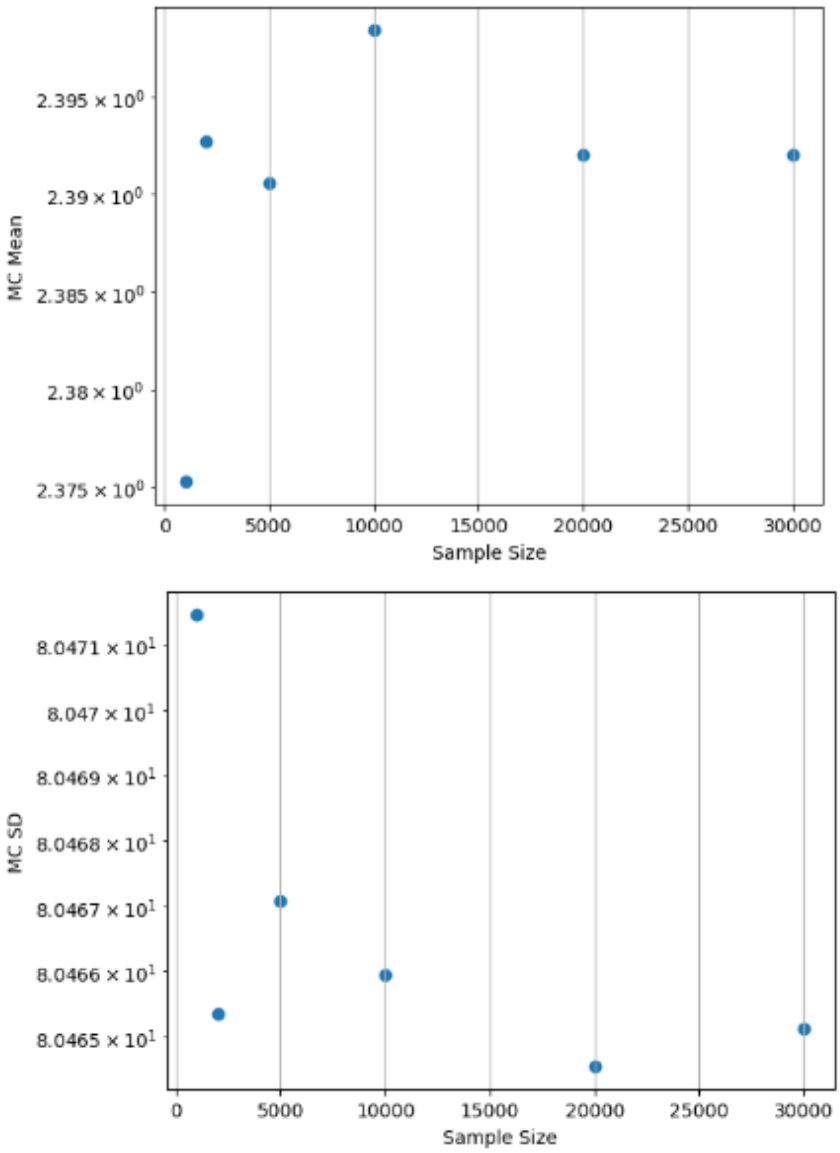

Figure 3.14: *Case 2: Results of MC using different sample sizes. Mean QoI value (top) and SD (bottom) are shown.*

The OLS PCE solution is also computed. Good quality results are obtained with $N = 75$ function calls, with a sampling ratio of $SR = 2.14$. The expansion approximated the MC moments with significant accuracy (Table 3.10), Compared to Case 1, a smaller number of samples is needed for MC convergence, as well as a smaller SR for an accurate OLS solution. This is explained by the fact that algebraically, this function turned out to be less complex than the borehole function, which was expected. The low non-linearity in combination with the lower dimensionality constitutes an easier PCE and UQ case.

The small percentage of high-magnitude coefficients in the OLS solution (Fig. 3.21) implies that the QoI function is moderately non-linear and a small set of terms

Table 3.10: *Case 2: OLS PCE Results.*

Statistic	OLS Prediction	MC
Mean	80.47456	80.46513
SD	2.36943	2.39540
MARE	0.00149	—

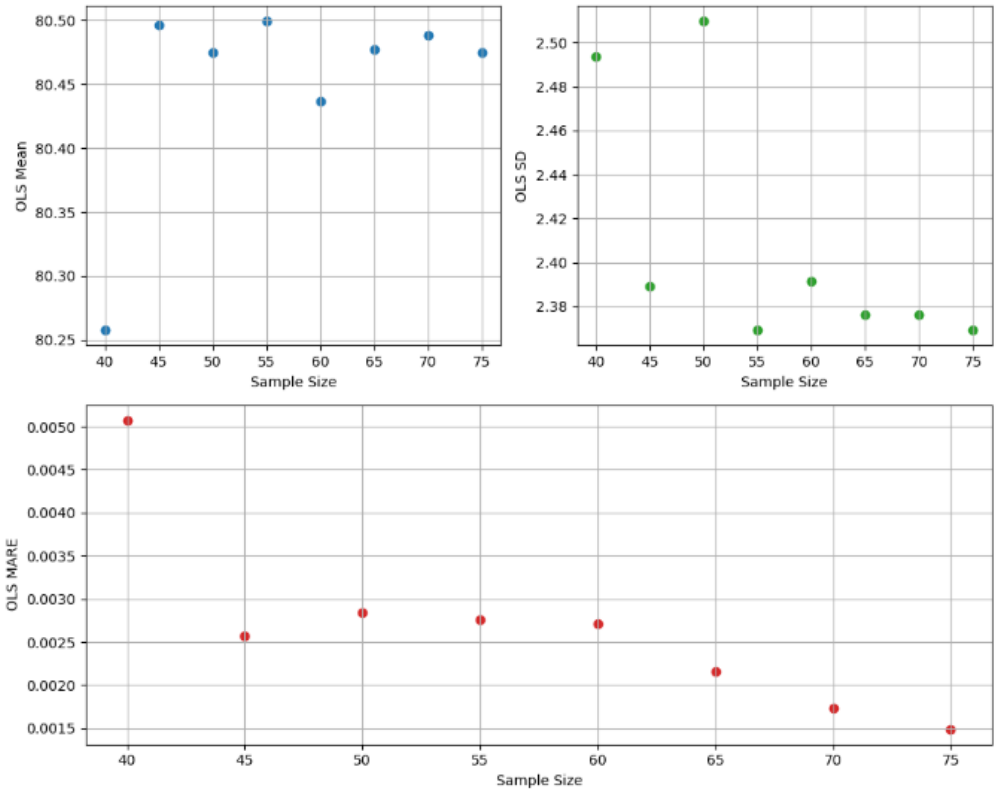

Figure 3.15: *Case 2: Computed Values of the OLS PCE Mean (top-left), SD (top-right), and MARE (bottom), using the OLS method, for various numbers of N samples (function calls).*

Table 3.11: *Case 2: Dominant PCE coefficients.*

PCE Coefficient	Multi-index	Variable
80.47456	(0,0,0,0)	Constant
2.32334	(0,0,0,1)	α
0.35359	(0,0,1,0)	ρ
-0.22734	(0,1,0,0)	μ
-0.09439	(0,0,2,0)	ρ^2

is important. If only the terms of Table 3.11 are used in the PCE basis, the SD drops from 2.36943 to 2.36295, remaining close to the MC estimate. Therefore, the

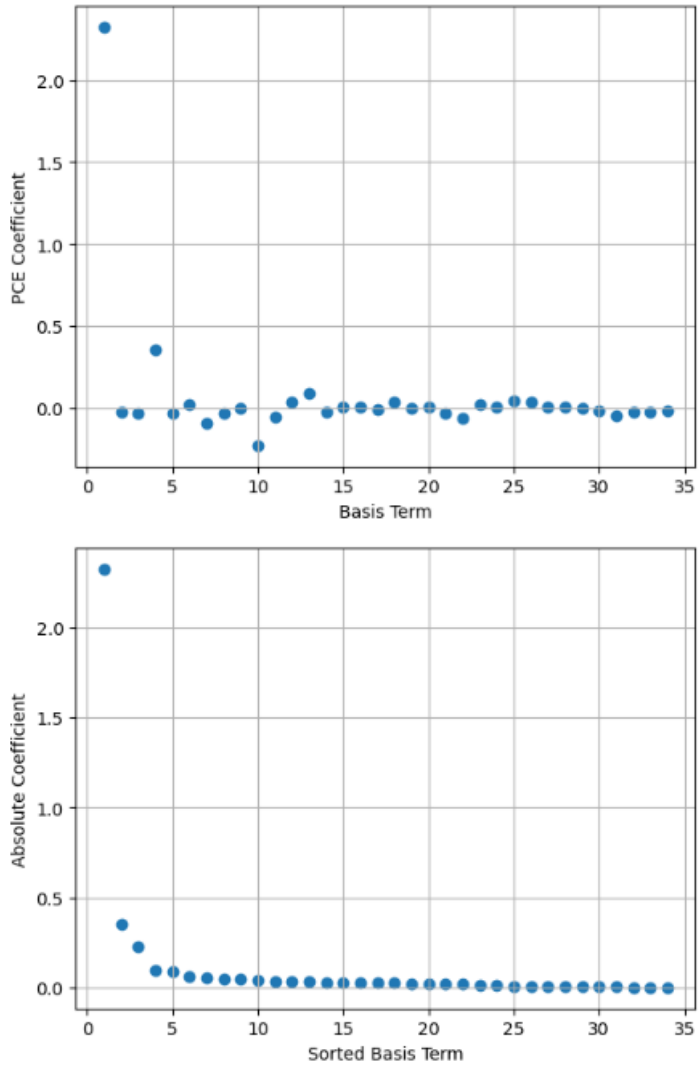

Figure 3.16: *Case 2: OLS Solution using $N = 75$ evaluations (0th term omitted). The solution coefficients (top) and their sorted absolute values (bottom) are illustrated.*

use of sparse methods is likely to produce results with acceptable accuracy using undersampled sets.

As far as the dominant terms are concerned, it is expected for the angle of attack to have a significant influence on the aerodynamic performance, even with such small perturbations. This is evident if the C_L/C_D polar is examined (Fig. 3.17. On the contrary, freestream velocity does not significantly influence the uncertainty of aerodynamic efficiency, since in the given range of values, the Reynolds number is not affected in the same magnitude as it is changed by perturbations in density and velocity. Density and viscosity are important, with density obtaining a larger PCE coefficient in its univariate linear term. This is why its quadratic term also comes up in the dominant terms, showing that the Reynolds number has a non-linear influence in aerodynamic efficiency.

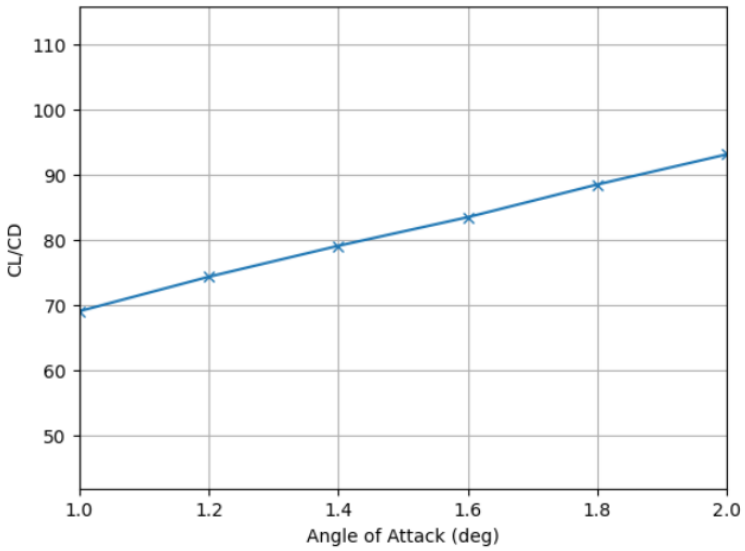

Figure 3.17: *C_L/C_D polar*

.

3.2.4 Use of Sparse Methods

With the conclusions of Case 1, the different sparse methods are implemented for different sizes of LHS sets. Sample sizes of size $N = 10, 12, 14, .., , 34$ are used. Since it is important to not only examine the sparse PCE methods' accuracy, but also show their efficiency in reducing sample size, OLS expansions derived from moderately oversampled sets of 40 ($SR = 1.11$) and 50 samples ($SR = 1.43$ are produced. With the SR not having reached the empirical range it is possible for sparse expansions to outperform OLS expansions that use more function calls.

General Observations

Compared with Case 1, this UQ example proved less challenging. Before the use of sparse methods, there were several indications that show that Case 2 is governed by a less complex function than Case 1. Firstly, both the MC simulation and the OLS solution required fewer samples. The OLS solution has even less important terms than that of Case 1 (Fig. 3.21) and its quality in predicting statistical moments and test samples is significantly lower than those of the previous benchmark.

The lower difficulty of this regression problem is also present in the sparse PCE results (Tables 3.12, 3.13). Sparse expansions accurately predict the mean, SD and the test set samples, even in severely undersampled regimes. The easiest UQ goal is the prediction of mean and the hardest is SD prediction. Sparse results demonstrate sample economy, where undersampled sets produce expansions, which occasionally outperform moderately oversampled PCEs and approximate the MC and benchmark results (Figures 3.19, 3.20).

Table 3.12: *Case 2: Sparse PCE predictions of Mean value $N = 10, 20, 30$ function calls. OLS solutions with $N = 40, 50$ samples are also reported for comparison.*

Method	$N = 10$	$N = 20$	$N = 30$	$N = 40$ **(OLS)**	$N = 50$ **(OLS)**
OMP	80.533334	80.430529	80.485890	—	—
LASSO	80.160669	80.431773	80.426907	—	—
rLASSO	80.418796	80.440956	80.472847	—	—
pLASSO	80.438995	80.445114	80.472848	—	—
OLS				80.257755	80.496018
MC Mean: 80.46513					

Table 3.13: *Case 2: Sparse PCE predictions of SD $N = 10, 20, 30$ function calls. OLS solutions with $N = 40, 50$ samples are also reported for comparison.*

Method	$N = 10$	$N = 20$	$N = 30$	$N = 40$ **(OLS)**	$N = 50$ **(OLS)**
OMP	2.36497	2.28434	2.42672	—	—
LASSO	1.92110	2.28960	2.31155	—	—
rLASSO	2.17705	2.29640	2.37353	—	—
pLASSO	2.19746	2.29997	2.37353	—	—
OLS				2.49357	2.38921
MC SD: 2.39540					

Prediction of Mean

While there are differences in the accuracy of mean prediction across different methods (OLS-sparse-reduced basis), even the highest deviations from the MC value are relatively small and this UQ goal was easily achieved. This is clear in Fig. 3.18. Standard LASSO illustrates the worst performance in the mean value metric. Sparse PCE expansions using more than 15 function calls ($N > 15$) estimate the mean value with greater accuracy than the OLS expansion using $N = 40$ samples, with OMP, pLASSO and rLASSO also outperforming the expansion using $N = 50$ in the highest sample range. This illustrates the capability of sparse expansions to achieve better or similar results to OLS based solutions using less samples.

SD and MARE

With all expansions approximating the mean value accurately, the calculation of SD and the prediction of the random test set serve as the main metrics that show PCE performance. For the prediction of SD, the worst performance is detected in the standard LASSO expansions, similar to Case 1, as seen in Fig. 3.19. In contrast with Case 1, the OMP algorithm is significantly more successful, showing accurate SD prediction across all the sample sizes. pLASSO and rLASSO show

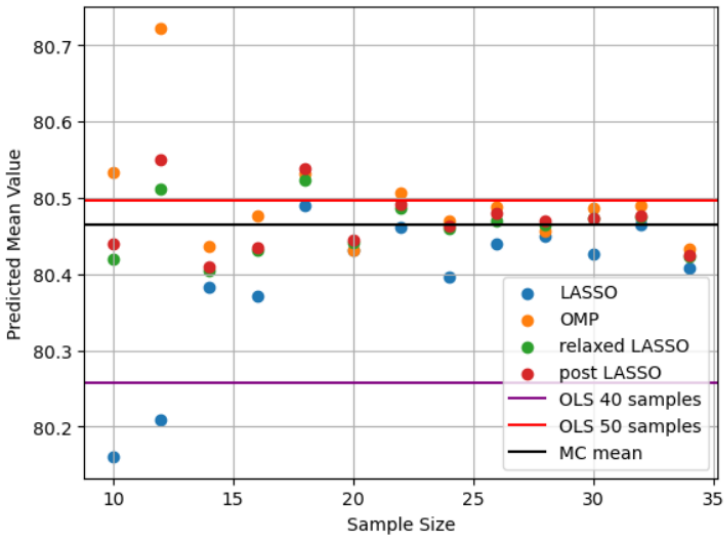

Figure 3.18: *Case 2: Predicted Mean Value across different sample sizes (N). Results for OLS with 40 samples (purple line) and 50 samples (red line), LASSO (blue dots), pLASSO (red dots), rLASSO (green dots), OMP (orange dots) are compared with the MC estimate (black line).*

similar results in SD, clearly outperforming standard LASSO, with their accuracy being comparable to that of OMP. SD predictions by OMP, pLASSO and rLASSO are better approximations to the MC estimate than the ones made by the OLS expansion using $N = 40$ samples for $N > 15$. While these values do not reach the accuracy of the predicted SD by the OLS expansion using $N = 50$ function calls, they remain significantly close to the MC value for $N < 20$ sample sizes. Observing the order of predicted values for each sample set, it is consistent with Case 1 results, with LASSO predictions obtaining the smallest values, OMP the highest and rLASSO and pLASSO predictions lying in between.

As far as test set prediction is concerned, low errors are present for all methods, highlighting how easier the regression problem of Case 2 is, compared with Case 1. It is also clear that undersampled sparse PCE expansions perform better in this task than moderately oversampled OLS-based expansions, showing their efficiency in evaluation costs.

Sparse PCE Coefficients

The coefficients of the sparse expansions using $N = 10, 20, 30$ samples are plotted with the OLS solution of $N = 75$ samples in Fig. 3.21. Unimportant features are generally not kept and therefore the expansions' accuracy is tested in the values of the important coefficients (Tables 3.14, 3.15, 3.16). The term that the sparse expansions are most frequently ignoring, is the one referring to the quadratic term of density (ρ^2). In terms of accuracy, OMP initially outperforms pLASSO and rLASSO for $N = 10$ and the opposite happens for $N = 30$, with standard LASSO

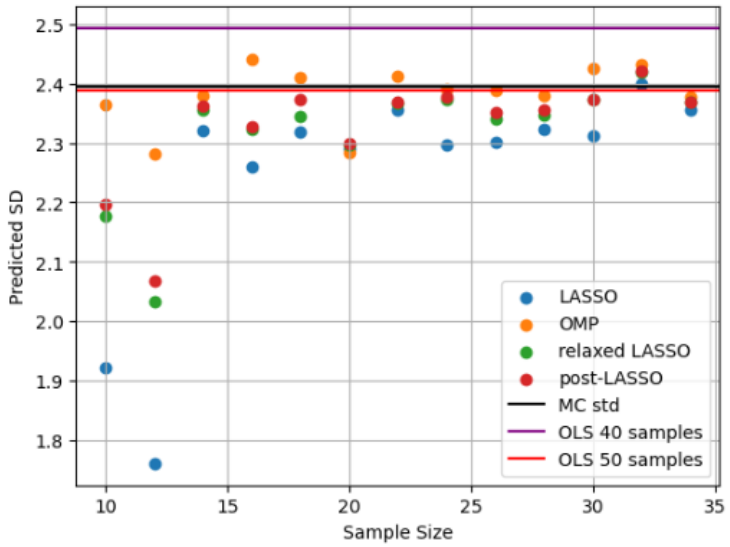

Figure 3.19: *Case 2: Predicted SD across different sample sizes (N). Results for OLS with 40 samples (purple line) and 50 samples (red line), LASSO (blue dots), pLASSO (red dots), rLASSO (green dots), OMP (orange dots) are compared with the MC estimate (black line).*

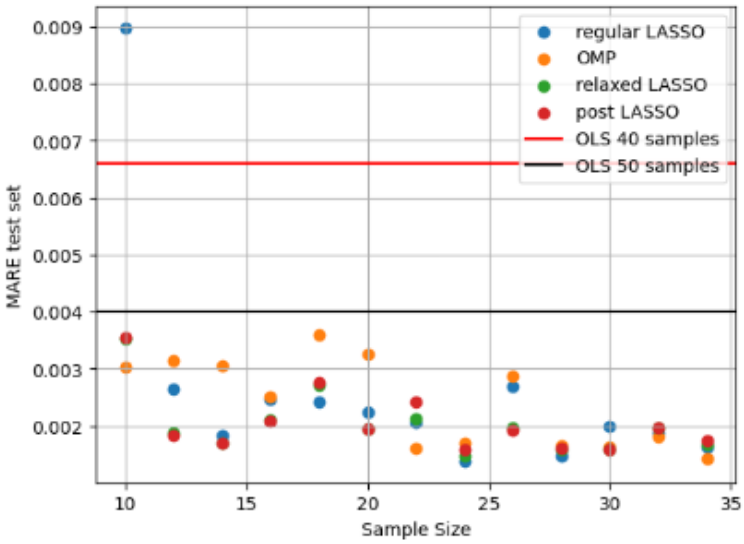

Figure 3.20: *Case 2: MARE of test set across different sample sizes. Results for OLS with 40 samples (red line) and 50 samples (black line), LASSO (blue dots), pLASSO (red dots), rLASSO (green dots), OMP (orange dots).*

having slightly worse performance.

3.2.5 Conclusions from Case 2

Compared with Case 1, the second case study proved to be a less challenging UQ problem, leading to improved performance of both OLS and sparse PCE approaches.

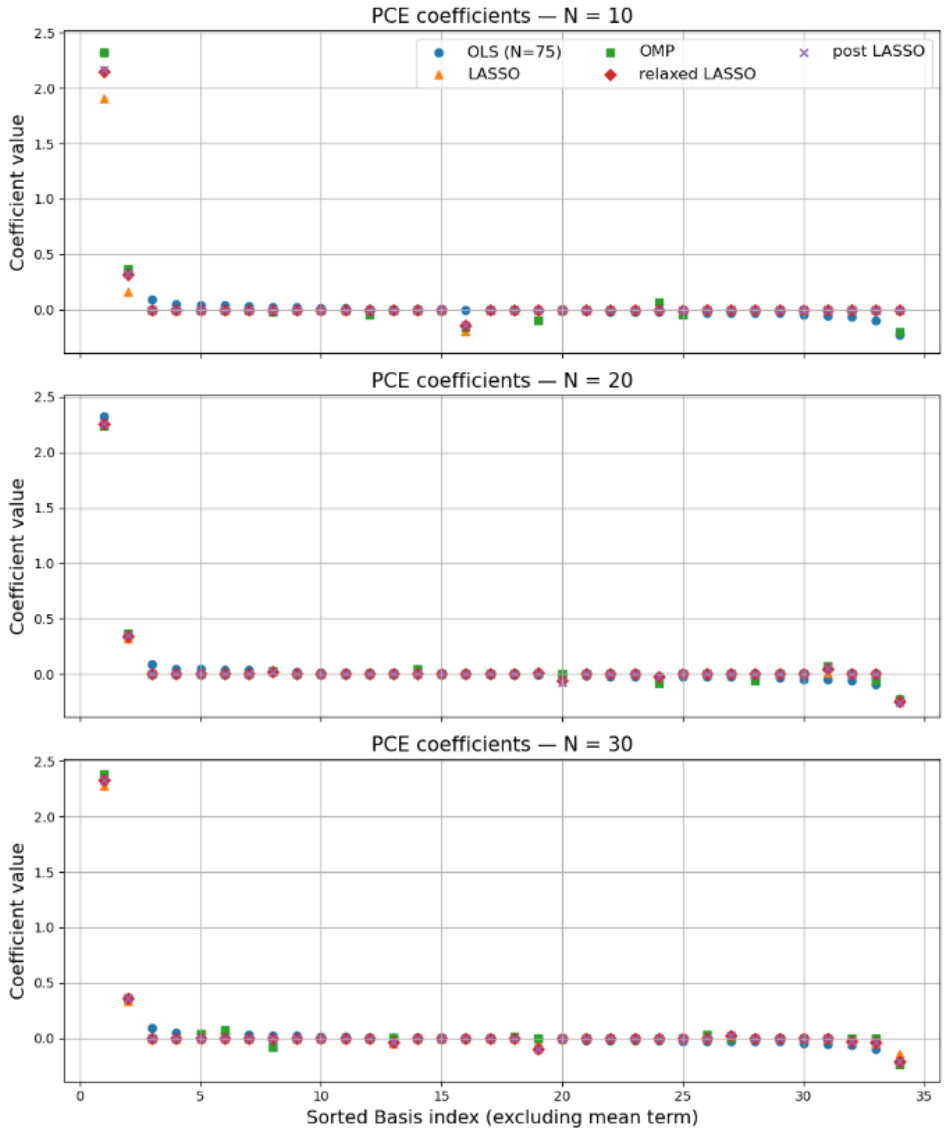

Figure 3.21: *Case 2: PCE coefficients from LH sets of size $N = 10$ (top), $N = 20$ (middle), and $N = 30$ (bottom). Solutions obtained from LASSO (orange), OMP (green), rLASSO (red) and pLASSO (purple) are compared to the OLS solution of $N = 75$ samples (blue). The basis terms are ordered so that the OLS expansion is sorted.*

The main conclusions are:

- The low dimensionality of the problem and the moderate non-linearity of the QoI enabled accurate approximations with fewer samples. MC convergence was reached with a relatively small number of evaluations, and OLS provided accurate results with only $N = 75$ function calls.

- The dominant PCE terms were mostly univariate, with angle of attack having the largest effect on aerodynamic efficiency, followed by viscosity and density.

Table 3.14: *Case 2: Dominant PCE coefficients for $N = 10$ samples.*

Variable	OLS (75)	LASSO	OMP	rLASSO	pLASSO
Constant	80.47456	80.16067	80.53333	80.41880	80.43900
α	2.32334	1.90435	2.31787	2.14953	2.16872
ρ	0.35359	0.16445	0.36582	0.31395	0.32564
μ	-0.22734	0.00000	-0.20293	0.00000	0.00000
ρ^2	-0.09439	0.00000	0.00000	0.00000	0.00000

Table 3.15: *Case 2: Dominant PCE coefficients for $N = 20$ samples.*

Variable	OLS (75)	LASSO	OMP	rLASSO	pLASSO
Constant	80.47456	80.43177	80.43053	80.44096	80.44511
α	2.32334	2.25536	2.23847	2.25638	2.25684
ρ	0.35359	0.32052	0.36309	0.33960	0.34823
μ	-0.22734	-0.22683	-0.23414	-0.24751	-0.25687
ρ^2	-0.09439	0.00000	-0.05589	0.00000	0.00000

Table 3.16: *Case 2: Dominant PCE coefficients for $N = 30$ samples.*

Variable	OLS (75)	LASSO	OMP	rLASSO	pLASSO
Constant	80.47456	80.42691	80.48589	80.47285	80.47285
α	2.32334	2.28133	2.38511	2.33326	2.33327
ρ	0.35359	0.33546	0.35773	0.36363	0.36363
μ	-0.22734	-0.14017	-0.23770	-0.20734	-0.20734
ρ^2	-0.09439	-0.04963	0.00000	-0.03811	-0.03812

In contrast, freestream velocity had negligible influence in the given range.

- Sparse PCE methods achieved lower test set errors than both OLS($N = 40$) and OLS($N = 50$), confirming their advantage in predictive accuracy despite using fewer samples.

- The prediction of the SD followed the same order as in Case 1: standard LASSO produced the lowest values, OMP the highest, and rLASSO/pLASSO lay in between. This consistency across cases highlights the characteristic biases of each method.

- OMP recovered compared to Case 1 and achieved accurate SD predictions, but pLASSO and rLASSO were again the most stable methods overall. They outperformed OLS($N = 40$) in mean and SD prediction and reached accuracy close to OLS($N = 50$), while also maintaining superior test set performance.

3.3 Case 3 - Airfoil Shape Uncertainty

3.3.1 Problem Formulation

The aerodynamic efficiency of the N16103 airfoil with geometric uncertainties was examined. With the Karhunen-Loève expansion the uncertainty was parametrized. Evaluating the eigenvalues of the covariance matrix, the first 18 terms of the expansion were able to retain 97% of the variance. Therefore, the uncertain geometry was parametrized to 18 KLE modes with uncertain amplitudes. All 18 of the uncertain variables are of Gaussian type:

- $\xi \in N[0, 0.3^2]$: KLE mode amplitude

The airfoil geometry is parametrized:

$$X(\mathbf{x}, \xi) = \sum_{i=1}^{18} \sqrt{\lambda_i} \phi_i(\mathbf{x}) \xi_i,$$

The **QoI** is again C_L/C_D. Setting 2 as maximum **Chaos Order** was sufficient in the highly non-linear example of Case 1 and a higher order significantly decreases the computational feasibility of this application due to both the computational cost of a CFD evaluation and the high input dimensionality. Via the total-order truncation formula the number of basis polynomials and unknown coefficients is 190.

3.3.2 Simulation Parameters

The flow around an isolated N16103 airfoil (Fig. 3.22) was simulated. The computations were carried out with the in–house code PUMA [3], which solves the compressible, Reynolds-Averaged Navier–Stokes equations. For the modelling of turbulence, the Spalart–Allmaras model is used in this case. A second-order Roe scheme is used for the spatial flux, while time marching employs a multi-stage Runge–Kutta scheme. Steady flow is assumed and pseudo-time marching occurs until convergence. The CFD simulations of Case 3 were performed on the NTUA GPU cluster equipped with 4 NVIDIA Tesla V100 GPUs. Each simulation required approximately 3 minutes of wall-clock time on a single V100 GPU.

Table 3.17: *Freestream and inlet boundary conditions (SI units).*

Quantity	**Value**	**Unit**
Total pressure, p_∞	21662.72	Pa
Total temperature, T_∞	216.65	K
Dynamic viscosity, μ_∞	2.39824e-5	Pa s
Angle of Attack, α	1.0	$^\circ$
Maximum CFL	10	(dimensionless)

The freestream Mach number is 0.86, which lies in the transonic region, so it is expected for shockwave formation to occur on the airfoil's suction side.

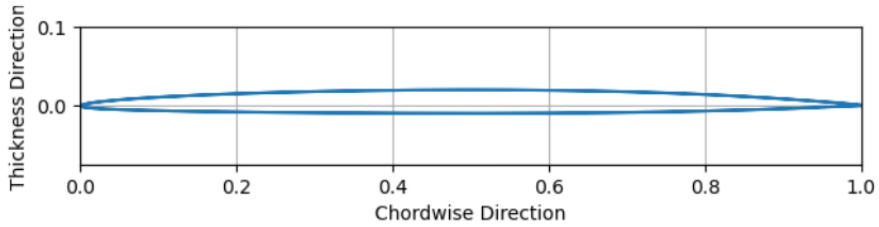

Figure 3.22: *Geometry of the N16103 airfoil*

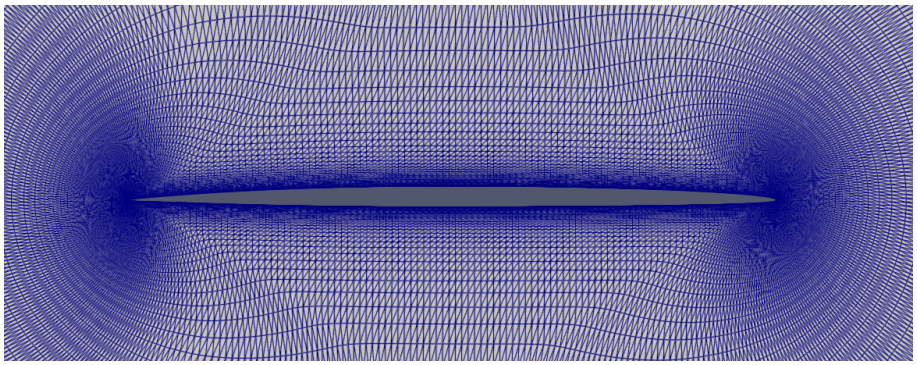

Figure 3.23: *Case 3: Grid in the airfoil region.*

3.3.3 Nominal Solution

Firstly, the flow around the undisturbed airfoil is examined. The sudden jump in pressure detected in the suction side (Fig. 3.24) denotes the existence of a shockwave. The output of the nominal simulation is presented (Table 3.18).

Table 3.18: *Nominal case aerodynamic forces and performance*

Quantity	**Value**
Lift (L)	2202.12406
Drag (D)	45.97910
Lift-to-Drag ($L/D = C_L/C_D$)	47.89402

3.3.4 OLS solution - Ground Truth

Due to the high computational cost of a CFD run and the high dimension of input variables, an MC simulation is infeasible. Hence, the only way to obtain the moments with which the sparse PCE will be evaluated is via an over-sampled Least-Squares expansion. By constantly increasing the oversampling factor, the moments and the prediction error to a randomly selected test set of 50 samples are evaluated. The

Figure 3.24: *Case 3: Pressure distribution of the Nominal Solution.*

sample size needed for an OLS solution was $N = 600$ ($SR = 3.15$). The mean and SD predicted by this expansion can be used as the ground truth. The need for high oversampling factors stems from the large number of basis functions and uncertain variables.

Table 3.19: *Oversampled PCE ($N = 600$) Results*

Statistic	**Value**
Mean	48.5462
SD	1.1658
MARE	0.0097

It is evident that the mean value is not equal to the output of the nominal case, which underlines that the mean value is important to accurately estimate for UQ.

The solution is characterized by rapid decrease in coefficient magnitude, similar to that of Case 2 so success in the implementation of similar methodologies is expected. The OLS solution if the 6 most dominant basis terms are kept yields an SD of 1.097, (-5.9% decrease from the "ground truth" value of Table 3.19).

As expected, the first KLE modes are more influential to the uncertainty of the aerodynamic efficiency. The importance of the first mode is such that its second order term and non-linear effects are more important than the first order effects of most modes. To understand how these eigen functions influence the aerodynamics, an exaggerated example of disturbed airfoils by each mode were compared to the baseline shape, along with an unimportant mode to illustrate the difference. The 11th mode was one of the least activated ones with its PC coefficients being orders of magnitude lower than those of the aforementioned terms. In Fig. 3.27 it is illustrated

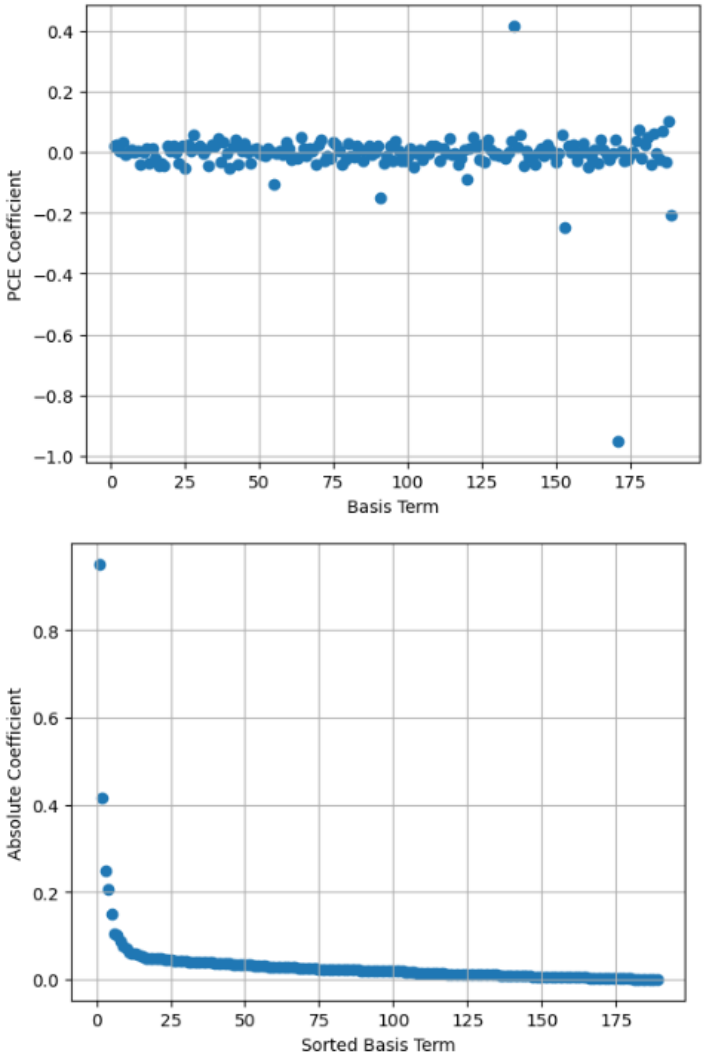

Figure 3.25: *Case 3: OLS Solution (0th term omitted). The solution coefficients (top) and their sorted absolute values (bottom) are illustrated.*

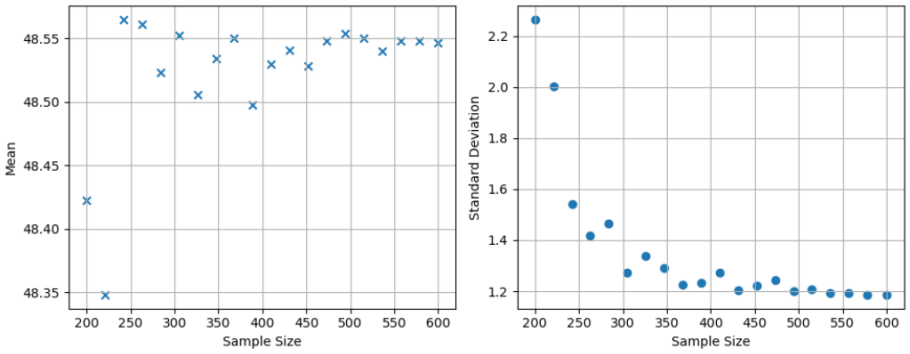

Figure 3.26: *Case 3: OLS PCE Mean (Left) and SD (Right)*

Table 3.20: *Case 3: Dominant PCE coefficients (OLS, $N = 600$). Order: Constant; ξ_1 (1st lin.); ξ_3 (3rd lin.); ξ_2 (2nd lin.); ξ_1^2 (1st quad.); ξ_6 (6th lin.).*

PCE Coefficient	Term	Interpretation
48.54616	Constant	Mean
-0.95106	ξ_1	1st KLE mode
0.41544	ξ_3	3rd KLE mode
-0.24811	ξ_2	2nd KLE mode
-0.20512	ξ_1^2	1st KLE mode (quadratic)
-0.15057	ξ_6	6th KLE mode

how the important modes influence large regions of the foil and especially the suction side where Lift is produced and shock waves occur. They are characterized by low spatial frequency in contrast with the unimportant 11th term. The odd shape of this term is explained by the fact that it expresses a less important base shape of the airfoil as far as covariance is concerned.

3.3.5 Use of Sparse Methods

Sparse PCE expansions are created with LH sample sets of size $N = 40, 60, 80, ..., 160$. Their performance in the three used metrics (Mean value, SD, MARE) is compared with the reference OLS solution ($N = 600$) as well as an OLS expansion using $N = 300$ samples. Firstly, the moments of the sparse expansions for sizes ($N = 40, 100, 160$) are presented (Tables. 3.21, 3.22).

Table 3.21: *Case 3: Sparse PCE predictions of Mean value with $N = 40, 100, 160$ function calls. OLS solution with $N = 300$ samples is also reported.*

Method	$N = 40$	$N = 100$	$N = 160$	**OLS** ($N = 300$)
LASSO	48.4224	48.4174	48.4763	48.5358
OMP	48.4382	48.4208	48.5260	
rLASSO	48.4406	48.4270	48.5126	
pLASSO	48.4502	48.4311	48.5170	
Reference OLS Mean ($N = 600$): 48.5462				

Mean Prediction

With evident differences in accuracy, significantly small deviations are present for all mean predictions, as seen in Fig. 3.28.

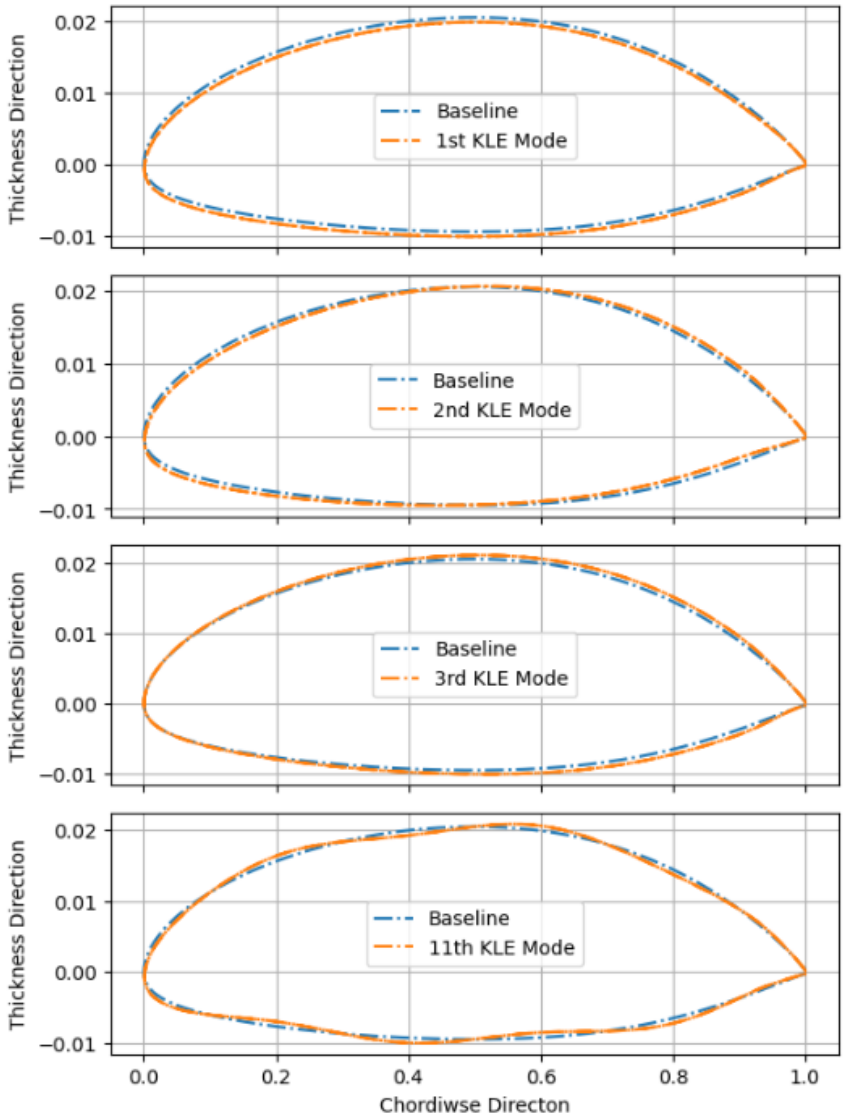

Figure 3.27: *Case 3, KLE modes visualization, 1st (top), 2nd (second), 3rd (third) and 11th (bottom), axes not in scale*

Table 3.22: *Case 3: Sparse PCE predictions of SD with* $N = 40, 100, 160$ *function calls. OLS solution with* $N = 300$ *samples is also reported.*

Method	$N = 40$	$N = 100$	$N = 160$	**OLS** ($N = 300$)
LASSO	0.9194	1.0318	0.9890	1.3182
OMP	1.3522	1.2974	1.2462	
rLASSO	1.0223	1.1930	1.1439	
pLASSO	1.0741	1.2280	1.1643	
Reference OLS SD ($N = 600$)**: 1.1658**				

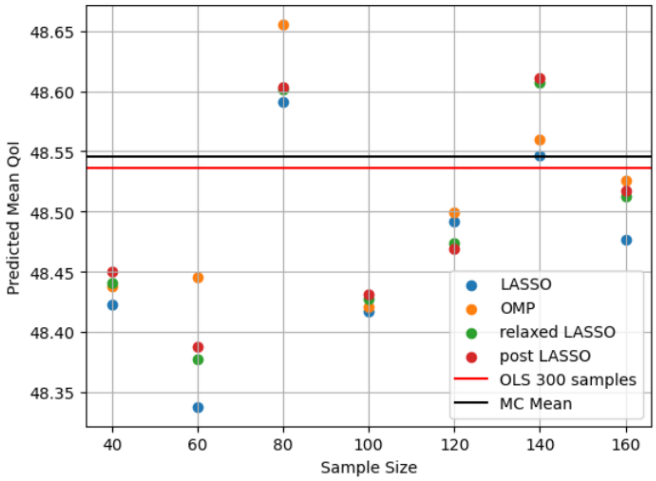

Figure 3.28: *Case 3: Absolute Relative Deviation of predicted Mean from the MC value across different sample sizes. Results for OLS with 300 samples (black line), LASSO (blue dots), pLASSO (red dots), rLASSO (green dots), OMP (orange dots).*

SD and MARE

The differences in test set predictions emulate the results from Case 1, shown in Fig. 3.30. Standard LASSO expansions obtain the lowest MARE values, with pLASSO and rLASSO showing slightly worse predictions. OMP is consistently the worst method in the MARE metric. Undersampled sparse expansions consistently exhibit lower prediction errors than the OLS expansion that processes $N = 300$ samples ($SR = 1.58$).

Test Set Prediction

The predicted test set values for the expansions with sample size $N = 100$ ($SR = 0.53$) and the OLS expansion with $N = 300$ are examined in Fig. 3.31. The worst prediction is seen by the moderately oversampled OLS expansion and the best by LASSO-based solvers, in accordance to the MARE results of Fig. 3.30.

sparse PCE Coefficients

In Case 3, the comparison of dominant PCE coefficients with the OLS ($N = 600$) reference (Tables 3.23, 3.24, and 3.25) shows that all sparse methods are able to recover the most important terms with satisfactory accuracy, even when the number of samples is severely limited. The constant term is predicted almost exactly across all sample sizes, while for the linear terms the same pattern observed in the previous cases is repeated: LASSO systematically underestimates the coefficients, OMP tends to overestimate them, and rLASSO and pLASSO provide values in between. The quadratic term ξ_1^2 exhibits larger differences, particularly for $N = 40$, where OMP significantly overshoots, but the accuracy improves as the sample size increases. At $N = 160$, the coefficients obtained by sparse methods are very close to those of

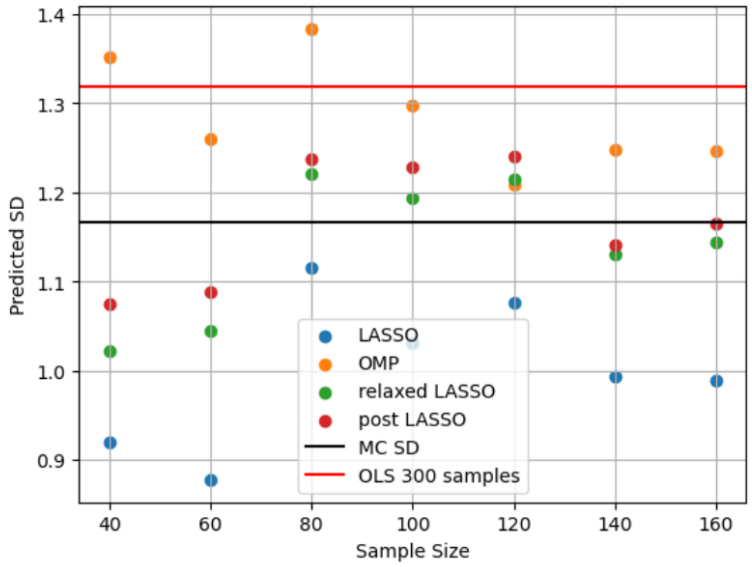

Figure 3.29: *Case 3: Absolute Relative Deviation of predicted SD from the MC value across different sample sizes. Results for OLS with 300 samples (black line), LASSO (blue dots), pLASSO (red dots), rLASSO (green dots), OMP (orange dots).*

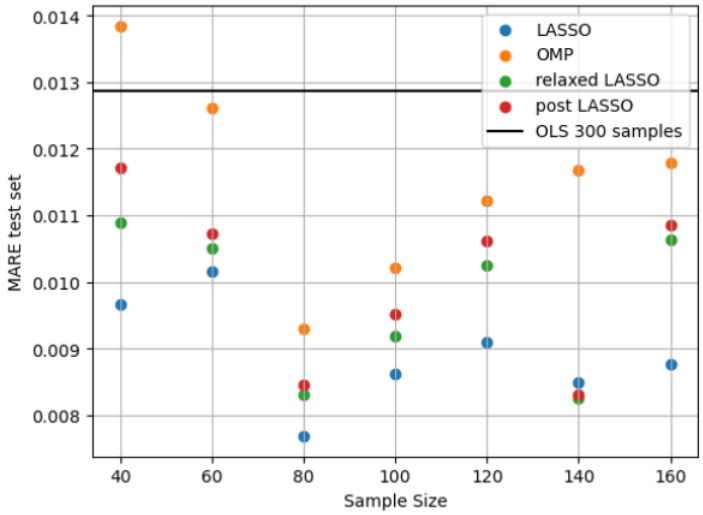

Figure 3.30: *Case 3: MARE of test set across different sample sizes. Results for OLS with 300 samples (black line), LASSO (blue dots), pLASSO (red dots), rLASSO (green dots), OMP (orange dots).*

OLS ($N = 600$), confirming the convergence of the expansions. Overall, rLASSO and pLASSO again demonstrate the most reliable and balanced performance, while OMP and standard LASSO retain their characteristic weaknesses.

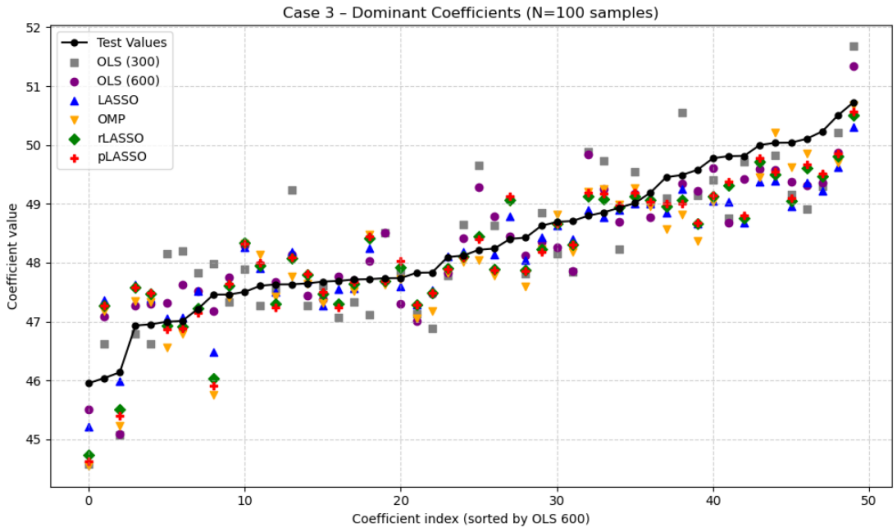

Figure 3.31: *Case 3: Predicted QoI values by expansions, using $N = 100$ function calls. Predictions obtained from LASSO (blue), OMP (yellow), rLASSO (green) and pLASSO (red) are compared to the sorted test QoI values (black) and the OLS predictions of $N = 300$ (grey) and $N = 600$ (purple) samples.*

Table 3.23: *Case 3: Dominant PCE coefficients for $N = 40$ samples. OLS($N = 600$) serves as reference.*

Term (order)	OLS (600)	LASSO	OMP	rLASSO	pLASSO
Constant	48.54616	48.42241	48.43823	48.44061	48.45019
ξ_1	-0.95106	-0.82200	-1.05467	-0.88847	-0.92034
ξ_3	0.41544	0.28022	0.29532	0.26702	0.24668
ξ_2	-0.24811	-0.14332	-0.37963	-0.13694	-0.11918
ξ_1^2	-0.20512	-0.16849	-0.40351	-0.19680	-0.20769
ξ_6	-0.15057	0.00000	0.00000	0.00000	0.00000

Table 3.24: *Case 3: Dominant PCE coefficients for $N = 100$ samples. OLS($N = 600$) serves as reference.*

Term (order)	OLS (600)	LASSO	OMP	rLASSO	pLASSO
Constant	48.54616	48.41744	48.42083	48.42700	48.43109
ξ_1	-0.95106	-0.84683	-0.97869	-0.92078	-0.93242
ξ_3	0.41544	0.37192	0.43075	0.40893	0.41930
ξ_2	-0.24811	-0.25417	-0.34535	-0.28895	-0.29663
ξ_1^2	-0.20512	-0.26565	-0.37152	-0.35378	-0.37233
ξ_6	-0.15057	-0.05810	-0.11730	-0.09876	-0.10649

3.3.6 Conclusions from Case 3

The analysis of Case 3, which involved uncertainty propagation with $d = 18$ KLE variables and costly CFD evaluations, demonstrates the following:

Table 3.25: *Case 3: Dominant PCE coefficients for $N = 160$ samples. OLS($N = 600$) serves as reference.*

Term (order)	OLS (600)	LASSO	OMP	rLASSO	pLASSO
Constant	48.54616	48.47627	48.52602	48.51257	48.51701
ξ_1	-0.95106	-0.89305	-1.02352	-0.95523	-0.96057
ξ_3	0.41544	0.29840	0.36052	0.36247	0.36938
ξ_2	-0.24811	-0.22429	-0.28803	-0.26280	-0.26708
ξ_1^2	-0.20512	-0.10192	-0.20725	-0.11079	-0.11162
ξ_6	-0.15057	-0.03121	-0.07526	-0.10436	-0.11214

- This is the first case where sparse PCE expansions provide a substantial practical benefit, since each CFD evaluation requires approximately 3 minutes. With $N = 100$ samples, rLASSO and pLASSO achieved more accurate SD predictions and MARE, and only a slightly underestimated mean compared to the OLS expansion with $N = 300$, corresponding to roughly 10 hours of saved computation.
- The overall behavior of the sparse methods resembles Case 1, in that OMP struggles under high input dimensionality, showing unstable performance and frequent overestimation of SD, while standard LASSO systematically underestimates the coefficients and remains insufficient. rLASSO and pLASSO again prove to be the most robust and balanced methods.
- At the same time, in terms of complexity, Case 3 is more similar to Case 2, since the results converge steadily as the sample size increases. For $N = 160$, all sparse expansions approach the reference OLS ($N = 600$) solution with high accuracy in both statistical moments and coefficients.
- In test set predictions, sparse expansions outperform the moderately oversampled OLS($N = 300$), confirming their efficiency in surrogate modeling tasks where predictive accuracy is essential.
- Coefficient analysis confirms that the first KLE modes dominate the uncertainty in aerodynamic efficiency. The non-linear influence of the first mode is strong enough that its quadratic term is more important than the first-order effects of many higher modes.

Chapter 4

Overall Conclusions

This thesis investigated the use of rPCE and sparse regression strategies for UQ in aerodynamic applications. The central aim was to assess the potential of sparsity-promoting methods to reduce the computational cost of UQ in fluid dynamics problems while maintaining accuracy. The analysis, carried out through progressively more challenging applications, has led to several important conclusions.

Firstly, it has been demonstrated that rPCE can approximate MC statistics with far fewer samples. OLS, while straightforward to implement, was found to provide stable solutions only under oversampling conditions, typically requiring the number of samples to exceed the basis size by a factor of about two. In square systems ($N = P$), OLS solutions proved inaccurate. In undersampled settings ($N < P$), the system becomes unsuitable for OLS. Sparse regression, on the other hand, is capable of exploiting the inherent compressibility of rPCE representations. In aerodynamic problems, only a limited subset of basis functions contributes significantly to the response, which means that sparse methods can identify and recover these dominant terms with substantially fewer samples. This compressibility underlies the efficiency of sparse regression and demonstrates that sparsity can be translated into sample economy without sacrificing accuracy.

The three applications examined in this work illustrate these conclusions from different perspectives. The first case, the classical borehole function, served as a benchmark in which the relative robustness of the methods could be directly compared. Here it became clear that while OLS is not robust under no not adequately oversampled sets, while sparse methods such as OMP and LASSO succeed in recovering the dominant terms of the expansion under moderate undersampling. The borehole function also revealed another important aspect: the dependence of regression quality on the sample set. Different LHS sets of the same size produced noticeably different results, underscoring the need for careful DoE when only limited data are available. Importantly, the benchmark also showed that reduced-basis approaches were the most stable, with pLASSO and rLASSO consistently outperforming LASSO and OMP and needing less samples to produce estimations of good quality,

The second case extended the investigation to an aerodynamic setting: the simulation of a NACA 2412 airfoil using XFOIL under uncertain flow conditions. This case provided the first aerodynamic application of sparse rPCE within this thesis. The results confirmed that sparse regression is able to capture the UQ statistics of the aerodynamic QoI with accuracy comparable to OLS, but at a significantly lower computational cost. This demonstrated the practical ability of sparse methods in realistic engineering problems, bridging the gap between benchmark functions and full CFD simulations.

The third case, involving a CFD simulation of a NACA 16103 airfoil with geometric uncertainty modeled via the KLE, represented the most computationally demanding test. In this setting, the benefits of sparse regression became even more apparent. By identifying only the important terms of the expansion, sparse methods rendered feasible an analysis that would otherwise have been prohibitively expensive. The findings of this case closely paralleled those of the borehole function, in the sense that dimensionality played a crucial role in determining robustness. At the same time, it became evident that geometric uncertainty introduces an additional layer of difficulty compared to flow uncertainty: the regression task is more challenging and requires higher dimensionality for accurate modeling, leading to a difficult PCE setting. Among the tested approaches, rLASSO and pLASSO consistently provided the most stable performance.

Taken together, these findings support a clear comparison between the studied methods. OLS remains a useful baseline, but its applicability is restricted to oversampled regimes and it fails completely in underdetermined settings. OMP offers efficiency and often accurate results when moderate undersampling is allowed, but its greedy nature limits its robustness. Standard LASSO is also prone to bad results in significantly undersampled applications. On the other hand, reduced-basis variants of LASSO, namely pLASSO and rLASSO, substantially improve performance by eliminating irrelevant terms before re-estimation; this held both in the Borehole benchmark and in the aerodynamic cases.

In conclusion, sparse regression rPCE methods, and especially reduced-basis LASSO variants, offer a powerful framework for UQ in aerodynamic problems. They enable the construction of accurate surrogate models under limited sample availability, which is critical when model evaluations are costly, as in CFD. Beyond the results of this thesis, future research should explore adaptive basis construction strategies, DoE techniques that consistently provide sample sets of good quality and extensions to three-dimensional CFD problems to further enhance efficiency and applicability in aerospace design and analysis.

Bibliography

[1] AllenZhu, Z., Gelashvili, R., Razenshteyn, I.: Restricted Isometry Property for General p-Norms. In: 31st International Symposium on Computational Geometry (SoCG 2015). Leibniz International Proceedings in Informatics (LIPIcs), vol. 34, pp. 451–460. Schloss Dagstuhl – Leibniz-Zentrum für Informatik (2015). https://doi.org/10.4230/LIPIcs.SOCG.2015.451

[2] Askey, R., Wilson, J.: Some basic hypergeometric orthogonal polynomials that generalize jacobi polynomials. In: Memoirs of the American Mathematical Society, vol. 54. American Mathematical Society (1985)

[3] Asouti, V., Trompoukis, X., Kampolis, I., Giannakoglou, K.: Unsteady CFD computations using vertex–centered finite volumes for unstructured grids on Graphics Processing Units. International Journal for Numerical Methods in Fluids **67**(2), 232–246 (May 2011)

[4] Bayram, I.: On the convergence of the iterative shrinkage/thresholding algorithm with a weakly convex penalty. IEEE Transactions on Signal Processing **64**(6), 1597–1608 (Mar 2016). https://doi.org/10.1109/tsp.2015.2502551, `http://dx.doi.org/10.1109/TSP.2015.2502551`

[5] Belloni, A., Chernozhukov, V.: Least squares after model selection in high-dimensional sparse models. Bernoulli **19**(2), 521–547 (2013)

[6] Berveiller, M., Sudret, B., Lemaire, M.: Stochastic finite element: a non intrusive approach by regression. European Journal of Computational Mechanics **15**(1–3), 81–92 (2006). https://doi.org/10.3166/remn.15.81-92

[7] Blatman, G., Sudret, B.: Adaptive sparse polynomial chaos expansion based on least angle regression. Journal of Computational Physics **230**(6), 2345–2367 (2011). https://doi.org/10.1016/j.jcp.2010.10.010

[8] Candes, E., Romberg, J., Tao, T.: Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. IEEE Transactions on Information Theory **52**(2), 489–509 (2006). https://doi.org/10.1109/TIT.2005.862083

[9] Chatzimanolakis, M., Kantarakias, K., Asouti, V.G., Giannakoglou, K.C.: Setting up the intrusive polynomial chaos method for uncertainty quantifica-

tion and adjoint-based optimization in compressible fluid flows. In: Proceedings of the Tenth International Conference on Computational Fluid Dynamics (ICCFD10). pp. 1–12. Barcelona, Spain (2018), https://www.iccfd.org/iccfd10/papers/ICCFD10-320-Paper.pdf, paper No. ICCFD10-2018-320

[10] Chen, S.S., Donoho, D.L., Saunders, M.A.: Atomic decomposition by basis pursuit. SIAM Journal on Scientific Computing **20**(1), 33–61 (1998)

[11] Diaz, P., Doostan, A., Hampton, J.: Sparse polynomial chaos expansions via compressed sensing and d-optimal design. Computer Methods in Applied Mechanics and Engineering **336**, 640–666 (2018). https://doi.org/10.1016/j.cma.2018.03.020, https://www.sciencedirect.com/science/article/pii/S0045782518301603

[12] Drela, M.: Xfoil: An analysis and design system for low reynolds number airfoils. Low Reynolds number aerodynamics (1989)

[13] Eldred, M.S.: Evaluation of non-intrusive approaches for wiener-askey polynomial chaos. Tech. Rep. SAND2008-5855, Sandia National Laboratories (2008). https://doi.org/10.2172/1145845, https://www.osti.gov/servlets/purl/1145845

[14] Eldred, M.S., Burkardt, J.: Comparison of non-intrusive polynomial chaos and stochastic collocation methods for uncertainty quantification. In: 44th AIAA Aerospace Sciences Meeting and Exhibit. No. 2009-0976, American Institute of Aeronautics and Astronautics (2009)

[15] Foucart, S., Rauhut, H.: A Mathematical Introduction to Compressive Sensing. Birkhäuser (2013)

[16] Friedman, J.H., Hastie, T., Tibshirani, R.: Regularization paths for generalized linear models via coordinate descent. Journal of Statistical Software **33**(1), 1–22 (2010). https://doi.org/10.18637/jss.v033.i01, https://www.jstatsoft.org/v33/i01

[17] Harper, W., Gupta, S.: Borehole function for uncertainty quantification. Journal of Hydrology (1983)

[18] Hosder, S., Walters, R., Perez, R.: A non-intrusive polynomial chaos method for uncertainty propagation in cfd simulations. AIAA Pap. **891** (01 2006). https://doi.org/10.2514/6.2006-891

[19] Judd, K.L., Maliar, L., Maliar, S., Valero, R.: Smolyak method for solving dynamic economic models: Lagrange interpolation, anisotropic grid and adaptive domain. Journal of Economic Dynamics & Control **44**, 92–123 (2014)

[20] Lüthen, N., Marelli, S., Sudret, B.: Sparse polynomial chaos expansions: Literature survey and benchmark. SIAM/ASA Journal on Uncertainty Quantification **9**(2), 593–649 (2021). https://doi.org/10.1137/20M1315774

[21] McKay, M.D., Beckman, R.J., Conover, W.J.: A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. Technometrics **21**(2), 239–245 (May 1979). https://doi.org/10.1080/00401706.1979.10489755

[22] Meinshausen, N.: Relaxed lasso. Computational Statistics & Data Analysis **52**(1), 374–393 (2007). https://doi.org/10.1016/j.csda.2006.12.019

[23] Moreau, J.J.: Proximité et dualité dans un espace hilbertien. Bulletin de la Société Mathématique de France **93**, 273–299 (1965)

[24] Orovic, I., Papic, V., Ioana, C., Li, X., Stankovic, S.: Compressive sensing in signal processing: Algorithms and transform domain formulations. Mathematical Problems in Engineering **2016**, 1–16 (01 2016). https://doi.org/10.1155/2016/7616393

[25] Pati, Y.C., Rezaiifar, R., Krishnaprasad, P.S.: Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In: Proceedings of the 27th Asilomar Conference on Signals, Systems and Computers. pp. 40–44. IEEE (1993)

[26] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, É.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research **12**, 2825–2830 (2011), http://jmlr.org/papers/v12/pedregosa11a.html

[27] Salehi, S., Raisee, M., Cervantes, M.J., Nourbakhsh, A.: Efficient uncertainty quantification of stochastic cfd problems using sparse polynomial chaos and compressed sensing. Computers Fluids **154**, 296–321 (2017). https://doi.org/https://doi.org/10.1016/j.compfluid.2017.06.016, https://www.sciencedirect.com/science/article/pii/S0045793017302281, iCCFD8

[28] Sudret, B.: Global sensitivity analysis using polynomial chaos expansions. Reliability Engineering & System Safety **93**(7), 964–979 (2008). https://doi.org/10.1016/j.ress.2007.04.002

[29] Tibshirani, R.: Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society: Series B (Methodological) **58**(1), 267–288 (1996)

[30] Tropp, J.A.: Greed is good: Algorithmic results for sparse approximation. IEEE Transactions on Information Theory **50**(10), 2231–2242 (2004). https://doi.org/10.1109/TIT.2004.836110

[31] Tropp, J.A., Gilbert, A.C.: Signal recovery from random measurements via orthogonal matching pursuit. IEEE Transactions on Information Theory **53**(12), 4655–4666 (Dec 2007). https://doi.org/10.1109/TIT.2007.909108, https://doi.org/10.1109/TIT.2007.909108

[32] Villete, S.: Advanced Techniques in Polynomial Chaos and Bayesian Inference for Uncertainty Quantification. Ph.d. thesis, National Technical University of Athens, Athens, Greece (2022)

[33] Wiener, N.: The homogeneous chaos. American Journal of Mathematics **60**(4), 897–926 (1938)

[34] Xiu, D., Hesthaven, J.S.: High-order collocation methods for differential equations with random inputs. SIAM Journal on Scientific Computing **27**(3), 1118–1139 (2005)

[35] Xiu, D., Karniadakis, G.E.: The wiener–askey polynomial chaos for stochastic differential equations. SIAM Journal on Scientific Computing **24**(2), 619–644 (2002)