ROBUST UNCERTAINTY QUANTIFICATION USING PRECONDITIONED LEAST-SQUARES POLYNOMIAL APPROXIMATIONS WITH l1-REGULARIZATION

J W van Langenhove, Didier D Lucor, A Belme

Robust uncertainty quantification using preconditioned least-squares polynomial approximations with ℓ_1-regularization *

J.W. Van Langenhove[†‡] D. Lucor[§] A. Belme[†‡]

2016

Abstract

We propose a non-iterative *robust* numerical method for the non-intrusive uncertainty quantification of multivariate stochastic problems with reasonably compressible polynomial representations. The approximation is robust to data outliers or noisy evaluations which do not fall under the regularity assumption of a stochastic truncation error but pertains to a more complete error model, capable of handling interpretations of physical/computational model (or measurement) errors. The method relies on the cross-validation of a pseudospectral projection of the response on generalized Polynomial Chaos approximation bases; this allows an initial model selection and assessment yielding a *preconditioned* response. We then apply a ℓ_1-penalized regression to the preconditioned response variable. Nonlinear test cases have shown this approximation to be more effective in reducing the effect of scattered data outliers than standard compressed sensing techniques and of comparable efficiency to iterated robust regression techniques.

1 Introduction

The aim of uncertainty quantification (UQ) comes down to numerically evaluating the *stochastic* response of some physical quantity of interest (QoI) $y(\boldsymbol{\xi})$, dependent upon various uncertain model parameters $\boldsymbol{\xi} := (\xi^{(1)}, \ldots, \xi^{(d)})$. Nowadays the majority of UQ developments makes use of *non-intrusive* uncertainty propagation techniques: i.e. the sampling of N model solutions is done using any legacy solver for the deterministic problem as a black box. When the number of uncertain input sources of a complex model is too large ($d \gg 1$), the efficient propagation of these uncertainties through the system remains an open problem due to the so-called curse of dimensionality. In any case, as the computational cost of a single simulation model is often high, e.g. for computational fluid dynamics simulations (CFD), a compulsory approach for UQ is to build a surrogate model $\tilde{y}$ that approximates the exact response of the QoI as accurately as possible based on the smallest number of *observations* or *samples* [3]. Once the samples are acquired, the construction and interrogation of the surrogate model itself should be computationally efficient so that the predictive capability of the metamodel is fully harnessed, for instance in terms of access to statistics of interest.

Different methods are available to construct these surrogate models, examples are Gaussian processes (Kriging) [49], Support Vector Machines [40], stochastic interpolation [43, 50] or stochastic spectral methods such as polynomial-based representations. In this paper we will focus on the use of the latter technique: generalized Polynomial Chaos (gPC) expansions [48, 6, 19, 51] that are well-suited for functions that belong to the ℓ_2 space, and may be seen as discrete least-squares projection on a polynomial space. The gPC represents a function as a weighted linear combination of P multivariate (polynomial) basis functions that are mutually orthogonal with respect to the probability measure of the uncertain parameters.

The coefficients $\boldsymbol{u}$ in the expansion may be computed in different ways: e.g. – using pseudospectral projection together with high-order (sparse) quadratures that are efficient for functions of moderately high dimensionality [37] or – by least-squares regression type of approach based on (random) data sampling. Written in a generic form, we have:

$$y \approx \tilde{y} = \hat{y}(\boldsymbol{\xi}) + e_T,$$

where, for a given model, $\tilde{y}$ represents some numerical simulations or "observations" of the system, $\hat{y}$ is what we call the surrogate model and e_T is the truncation error that obviously depends on $\boldsymbol{\xi}$. Because we deal with computer experiments, the observations are in practice corrupted by *model* errors. Let us consider a family of models characterized by unobserved random variables $\boldsymbol{\chi}$ (e.g. related to mesh quality criteria). Now, y may be modeled as a functional of $(\boldsymbol{\xi}, \boldsymbol{\chi})$ via a model error e_M, and we have:

$$y(\boldsymbol{\xi}, \boldsymbol{\chi}) = \tilde{y}(\boldsymbol{\xi}) + e_M, \quad \text{where } \tilde{y}(\boldsymbol{\xi}) = \hat{y}(\boldsymbol{\xi}) + e_T(\boldsymbol{\xi}).$$

*Reference: DOI: 10.1615/Int.J.UncertaintyQuantification.2016015915, International Journal for Uncertainty Quantification, Volume 6, Issue 1, 2016, pages 57-77

[†]Sorbonne Universités, UPMC Univ Paris 06, UMR 7190, Institut Jean le Rond d'Alembert, F-75005, Paris, France.

[‡]CNRS, UMR 7190, Institut Jean le Rond d'Alembert, F-75005, Paris, France.

[§]LIMSI, CNRS, Université Paris-Saclay, Campus universitaire bât 508, Rue John von Neumann, F-91405 Orsay cedex, France.

A frequent assumption found in the literature is the one of a stochastic noise model, where $e_M \equiv e_M(\boldsymbol{\chi})$ are centered i.i.d. (normal) random variables, with uniformly bounded variance. In real life, modeling errors are often biased, not uncorrelated and uniform (heteroscedasticity) and not normally distributed. Therefore, they bear a deterministic noise component that depends on $\boldsymbol{\xi}$ [10], such that $e_M \equiv e_M(\boldsymbol{\xi}, \boldsymbol{\chi})$. In practice, it is very hard to quantify these modeling errors as we do not directly observe $\boldsymbol{\chi}$. Moreover, the system may not be in the asymptotic regime for which we may have convergence information/estimates for the deterministic model error. The interplay between (the different scales related to) $\boldsymbol{\chi}$ and $\boldsymbol{\xi}$ is also very difficult to apprehend. Ideally, one would want to marginalize over the model error in order to assess the *conditional* random variable:

$$y(\boldsymbol{\xi}) := \mathbb{E}\left[y(\boldsymbol{\xi}, \boldsymbol{\chi}) \,|\, \boldsymbol{\xi}\right] = \hat{y}(\boldsymbol{\xi}) + e(\boldsymbol{\xi}), \quad \text{where } e(\boldsymbol{\xi}) = e_T(\boldsymbol{\xi}) + \mathbb{E}\left[e_M(\boldsymbol{\xi}, \boldsymbol{\chi}) \,|\, \boldsymbol{\xi}\right],$$

but this approach is often out of reach [22]. Another strategy is to perform the regression for a given model, which implicitly coincides with a given value of $\boldsymbol{\chi}$. When the dependence between the solution and the random parameters $\boldsymbol{\xi}$ is smooth, model errors are generally relatively independent from the parameters $\boldsymbol{\xi}$ and their effects often result in some form of biased predictions (e.g. under-prediction in case of model numerical diffusion). When the dependence is not smooth due to some brutal change in the solution physical regime, bifurcations, instabilities, transients, etc.[1], model or computational errors induce some local large-amplitude oscillations that may be seen as data *outliers* and are very detrimental to the stability of the stochastic quantification of the response. In this case, these large errors are unpredictably scattered and increasing the number N of samples of $\boldsymbol{\xi}$ does not help as the prediction essentially fits the model error. This is the well-known problem of *overfitting* when a model fits training data very well, but will do a poor job of predicting results for new data. A first step toward *robust* UQ in this framework would be to automatically detect the data outliers and reduce their influence in order to regularize the response on a given model basis. A simple way of thinking about the e_M error is to relate it, for instance, to the discretization error of the computational model. Indeed, it is not always possible due to prohibitive computational cost to lower the discretization error to levels that do not play a significant role in UQ. The case of CFD simulations of a compressible flow past an airfoil at random angle of attack presented later in the paper is a nice illustration of this setup and was the starting point of this work. For a given level of discretization (in practice a given mesh) and Mach number, a small variation in the angle of attack may induce a change in the flow from subsonic to transsonic regime, materialized by the emergence of a flow discontinuity (shock). This shock is then poorly captured if the mesh is not properly adapted in space, inducing large model errors and fluctuations in addition to the truncation error. In this case, the discretization error will strongly affect only few or a short range of the angle of attack realization values.
One of our contributions in this work is to propose a numerical approach that automatically detects data outliers and weighs them with low level of confidence. In this work, the detection and weighting is in part based on exhaustive surrogate model cross-validation namely the leave-one-out (LOO) technique. The LOO error estimation has been used before in the context of basis selection of gPC expansions [4, 25], and more generally in statistical learning theory for model selection [23]. The negative effects of the outliers on the construction of the surrogate model will then be minimized in order to avoid overfitting.

Another key aspect of this work is to take advantage of the sparsity of the solution structure. Indeed, the solution of high-dimensional problems is sometimes *sparse* (or near-sparse) at the stochastic level. This means that it may be accurately represented with only few terms when linearly expanded into a stochastic approximation space, such as the one encompassed by a gPC basis. In this case, the number s of dominant basis functions is small relative to the cardinality P of the full basis and the problem is said to be $s-$sparse. Promising approaches for solving this kind of problem involve compressed sensing (CS) techniques, also known under the names of Compressive Sensing, ℓ_1-minimization, convex relaxation and ℓ_1-regularized least-squares minimization. Relatively recent results in CS have made it clear that sparse functions can be accurately recovered from much fewer observations than necessary for classical solution methods [7, 9, 14]. Interestingly, this ability is preserved in the case of sparse solutions tainted by noise, as long as it is sufficiently regular and bears a low signal-to-noise ratio [8, 15, 18, 45].
Several research groups have recently been using CS in a gPC framework [16, 28, 52, 53] and have considered this noise as the truncation error of the gPC approximation. The efficiency of this approximation depends on the type and cardinality of the gPC approximation basis selected [4, 25] and the choice of the collocation samples to be used. The most readily available literature is about sparse Legendre and Hermite polynomials with *random* sampling. For both cases, different sampling strategies are possible: – standard sampling according to the underlying probability measure, and – asymptotic sampling according to the Chebyshev measure for Legendre polynomials, and to Hermite functions for Hermite polynomials [22, 42]. For s-sparse Legendre polynomial with maximal degree p, it was shown that the asymptotic relation between the number N of samples drawn according to Chebyshev distribution, s and p, guaranteeing recovery, is given by $N \asymp s \log^4(p)$ [36]. Chebyshev sampling has been shown to be superior to uniform sampling for elliptic stochastic partial differential equations of moderately high dimension ($d \sim 10$) [53], but the results can not be generalized. In fact, Yan et al. [52] show that for high-dimensional problems, sampling according to the Chebyshev measure can become less efficient. Interestingly, in case of standard sampling, the Chebyshev

[1] Or due to some soft system faults (e.g. bit-flips), nowadays more frequent in petascale high performance computing.

probability measure may be imposed afterwards by preconditioning the ℓ_1-minimization problem. This approach inspired us to use data-driven preconditioning to improve approximation robustness. Finally, a recently developed sampling strategy is the *coherence-optimal* sampling [22], which guarantees recoverability with a number of samples that is bound linearly by the number of basis functions up to a logarithmic factor.

Very recent works have investigated the efficiency of these methods for randomized *quadratures*: i.e. randomly subsampling among structured Gauss quadrature nodes [41, 54]. Using the bounds from [36], Tang & Iaccarino [41] show that for an efficient recovery of the gPC Legendre expansions, the number of observations scales with the sparsity s and only logarithmically with P. When the number of random dimensions is small to moderate, and more specifically when $p > d$, it is conceivable to directly rely on *complete* structured grids inherited from full or partial (also known as sparse) tensorization of quadrature rules, which is what we propose in our contribution. Moreover, the use of these regular grids minimizes *leverage* effects in regressions due to unusual design points.

The aim of this work is to fully harness the capability of CS techniques for UQ, even in the presence of scattered data outliers attributable to computational model errors that do not fall under the common Gaussianity and low signal-to-noise ratio assumptions. We wish to do so by regularizing the system response for a given computational model. We propose preconditioned compressed sensing in order to build robust polynomial surrogate of the stochastic response from sampling on structured grids. More specifically, after selecting the best model by cross-validation using numerical quadrature, the weight for each observation will be based on the inverse of its contribution to the cross-validation error of this model. Using confidence in samples in the form of a weighted least squares solution has been done before, see for example [55]. Here however, no a priori knowledge about the scale of the noise is needed, nor does one need to know beforehand which observations have been affected by the noise. This weighting of the observations can be used in combination with any available method for constructing the surrogate model. In this study, we have opted for the use of the Least Absolute Shrinkage and Selection Operator (LASSO) technique [44], which is known to be very robust, to compute the coefficients of the surrogate model, but as stated before, other methods can be used as well.

The structure of the paper is as follows: section 2 will briefly recall the key points of the collocated stochastic spectral approximation framework with and without ℓ_1-regularization. This will serve mostly as an introduction for our notations. In section 3, we will discuss how we derive observation weights using cross-validation and how it is interwoven with the ℓ_1-minimization constraint. The proposed technique will be demonstrated on several test problems in section 4. This paper ends with some conclusions and perspectives for future work.

2 Different formulations for the generalized Polynomial Chaos approximation

As stated in the introduction, gPC expansions will be used to express the surrogate model in a closed form [19, 27, 51]. Let $(\Omega, \mathcal{B}, \mathcal{P})$ be the probability space where Ω is the space of random events ω, this domain has a σ-algebra $\mathcal{B}$ and is equipped with a probability measure $\mathcal{P}$. The vector of random parameters can be written as $\boldsymbol{\xi} \equiv \boldsymbol{\xi}(\omega) = (\xi^{(1)}, \dots, \xi^{(d)})$, but we will often omit the dependence on ω to simplify notation. If we consider a $d-$variate functional $y : \mathcal{I}_{\boldsymbol{\xi}} \subseteq \mathbb{R}^d \to \mathbb{R}$, then any second-order random variable[2] $y(\boldsymbol{\xi}) \in \ell_2(\Omega, \mathcal{B}, \mathcal{P})$, can be expressed as a gPC expansion [51]:

$$y(\boldsymbol{\xi}) = \sum_{j=0}^{\infty} u_j \psi_{\mathrm{j}}(\boldsymbol{\xi}), \tag{1}$$

where $\psi_{\mathrm{j}}(\boldsymbol{\xi}) = \prod_{i=1}^{d} \psi_j^{(i)}(\xi^{(i)})$ are the multivariate basis functions that form a complete basis, orthonormal with respect to the probability measure $\rho_{\boldsymbol{\xi}}$ of the random input, and $\psi_j^{(i)}$ are the univariate basis functions along the i^{th} dimension. We assume that all $\xi^{(i)}$ are independent and thus $\rho_{\boldsymbol{\xi}} = \prod_{i=1}^{d} \rho^{(i)}(\xi^{(i)})$. Note that Ω is a Hilbert space and that we can write its inner product in terms of the expectation operator $< y, g > \equiv \mathbb{E}[y \cdot g]$, in this case:

$$\mathbb{E}\left[y(\boldsymbol{\xi}) g(\boldsymbol{\xi})\right] = \int_{\mathcal{I}_{\boldsymbol{\xi}}} y(\boldsymbol{\xi}) g(\boldsymbol{\xi}) \rho_{\boldsymbol{\xi}} d\boldsymbol{\xi}. \tag{2}$$

Instead of indexing the expansion of equation (1) on a single integer amounting to the cardinality of the entire approximation space, one can also make use of a multi-index notation that is equivalent. If Λ_p is an index set (to be defined) for multi-index $\boldsymbol{\gamma} = (\gamma_1, \dots, \gamma_d) \in \mathbb{N}_0^d$, then $\mathbb{P}_{\Lambda_p} \equiv \text{span}\{\psi_{\boldsymbol{\gamma}} \mid \gamma \in \Lambda_p\}$ and we can then write $\psi_{\boldsymbol{\gamma}}(\boldsymbol{\xi}) = \prod_{i=1}^{d} \psi_{\gamma_i}^{(i)}(\xi^{(i)})$ where $\psi_{\gamma_i}^{(i)}$ is the γ_i^{th} order basis function in dimension (i). Using the

[2] Here, we drop the tilde notation introduced in the introduction for simplicity.

notation introduced above, one can write out the truncated gPC expansion approximating y as follows:

$$y(\boldsymbol{\xi}) = \sum_{\boldsymbol{\gamma}\in\Lambda_p} u_{\boldsymbol{\gamma}}\psi_{\boldsymbol{\gamma}}(\boldsymbol{\xi}) + e_T(\boldsymbol{\xi}), \tag{3}$$

where $u_{\boldsymbol{\gamma}}$ are the coefficients corresponding to the $\psi_{\boldsymbol{\gamma}}$ basis[3]. We will restrict ourselves to tensor-product polynomial spaces $\mathbb{P}_{\Lambda_p}$, where Λ_p is an index set of "degree" p, and where $P = \dim(\mathbb{P}_{\Lambda_p}) \equiv \#\Lambda_p$, will denote the cardinality of the selected polynomial space. There are different ways of constructing the approximating polynomial spaces that will impact their cardinality:

- Tensor Product (TP): $\mathbb{P}^{\mathrm{TP}}_{\Lambda_p}$ with index set $\Lambda^{\mathrm{TP}}_p = \{\boldsymbol{\gamma} \in \mathbb{N}^d_0 : ||\boldsymbol{\gamma}||_\infty \leq p\}$,
- Total Degree (TD): $\mathbb{P}^{\mathrm{TD}}_{\Lambda_p}$ with index set $\Lambda^{\mathrm{TD}}_p = \{\boldsymbol{\gamma} \in \mathbb{N}^d_0 : ||\boldsymbol{\gamma}||_1 \leq p\}$, or
- Hyperbolic Cross (HC): $\mathbb{P}^{\mathrm{HC}}_{\Lambda_p}$ with index set $\Lambda^{\mathrm{HC}}_p = \{\boldsymbol{\gamma} \in \mathbb{N}^d_0 : \prod_{i=1}^d(\gamma_i + 1) \leq p + 1\}$ [39].

In this paper, without any loss of generality, we will be using approximation spaces of total degree (TD), so Λ_p will refer to Λ^{TD}_p in the following.

2.1 Galerkin projection

The first way of determining the coefficients is by use of a Galerkin projection [19]. One can write

$$\mathbb{E}\left[\sum_{\boldsymbol{\gamma}\in\Lambda_p} u_{\boldsymbol{\gamma}}\psi_{\boldsymbol{\gamma}}\psi_{\boldsymbol{\beta}}\right] = \mathbb{E}\left[\psi_{\boldsymbol{\beta}} y\right], \quad \forall\ \boldsymbol{\beta} \in \Lambda_p. \tag{4}$$

Here, with some abuse of notation, we have written the above equation as an equality instead of an approximation, the same will be done in the rest of this paper. Assuming the basis is *orthonormal*, the coefficients can be found simply by computing:

$$u_{\boldsymbol{\gamma}} = \mathbb{E}\left[\psi_{\boldsymbol{\gamma}} y\right] \quad \text{with} \ \ \boldsymbol{\gamma} \in \Lambda_p, \tag{5}$$

making use of a quadrature in the case of a pseudospectral implementation [27].

2.2 Least-Squares Minimization

One can also use linear regression to compute the unknown coefficients $u_{\boldsymbol{\gamma}}$, e.g. [11, 2]. The Least-Squares (LS) solution minimizes the residuals, $\boldsymbol{r} \equiv \boldsymbol{y} - \boldsymbol{\psi}_{\Lambda_p}\boldsymbol{u}$ in the ℓ_2-norm and may be written as an optimization problem:

$$\boldsymbol{u} = \underset{\mathbf{u}\in\mathbb{R}^P}{\operatorname{argmin}} \|\boldsymbol{y} - \boldsymbol{\psi}_{\Lambda_p}\mathbf{u}\|_2, \tag{6}$$

where $\boldsymbol{\psi}_{\Lambda_p}$ is the measurement matrix corresponding to the gPC expansion in the index set Λ_p. The solution to (6) is obtained by computing the following system written in matrix form:

$$\boldsymbol{u} = \left(\boldsymbol{\Psi}^{\mathsf{T}}_{\Lambda_p}\boldsymbol{\Psi}_{\Lambda_p}\right)^{-1}\boldsymbol{\Psi}^{\mathsf{T}}_{\Lambda_p}\mathbf{y}, \tag{7}$$

where $\mathbf{y}$ is a vector of observations of size $N\times 1$, $\boldsymbol{\Psi}_{\Lambda_p}$ the *measurement matrix* of size $N\times P$ with $\Psi_{ij} = \psi_{\mathrm{j}}(\boldsymbol{\xi_i})$, and $\boldsymbol{u}$ the vector of coefficients of size $P\times 1$. There has been a growing interest in understanding the conditions under which problem (6) leads to accurate and stable (multivariate) polynomial chaos approximations for data *randomly* and *independently* sampled according (or not) to their natural orthogonality measures [12, 30, 21]. More specifically, these studies focussed on the relation between the required number of samples and the cardinality of the approximation basis for different sampling measures. If one uses enough sampling points to be able to properly recover the orthonormality of the basis functions ψ_{j}, then the matrix $\left(\boldsymbol{\Psi}^{\mathsf{T}}_{\Lambda_p}\boldsymbol{\Psi}_{\Lambda_p}\right)$ is the identity matrix and the link between (7) and (5) becomes clear. The aforementioned works mainly deal with *noiseless* evaluations of the target function, and only few papers consider noisy data samples [29]. In any case, response fittings based on standard or *ordinary* LS type objective functions are not robust against outliers, i.e. data samples that strongly deviate from assumptions (e.g. of normality). It is said that the LS estimator has a breakdown point of $1/N$ because just one leverage point may cause it to break down [24]. In this case, the approximation might be biased, with an artificially inflated variance.

[3]If the functional to approximate is a random process, it may also depend on space and time and in that case the gPC coefficients will be deterministic space- and time-dependent fields.

2.2.1 Robust regression

In robust statistics, robust regression is a form of analysis designed to circumvent some limitations of traditional parametric and non-parametric methods by dampening the influence of outlying cases [24]. Most common robust regression methods fall into the class of $M-$estimators[4] which attempt to minimize the sum of a chosen *objective* (also called *loss*) function[5] of the residuals, i.e. $\sum_{j=1}^{N} \rho(\boldsymbol{r}_j)$. This minimization may be equivalently written as a *weighted* LS problem; the weight of each sample being expressed via the *score* function $\upsilon(\boldsymbol{r}) \equiv \partial\rho/\partial\boldsymbol{r}$, i.e. a derivative of the objective function at that point. Because of their connection to the residual values, the weights are iteratively evaluated until numerical convergence. In this framework, iteratively re-weighted least square (IRLS) algorithms are implemented for different choices of objective functions, e.g. leading to Huber , Tukey's *bisquare*, or Cauchy estimators, etc... [20, 26]. In our numerical applications, we will often derive our sample weights from the Cauchy objective function: $\rho(\boldsymbol{r}_j) = \log(1 + \boldsymbol{r}_j^2)/2$. The different $M-$estimators are influenced by the *scale* of the residuals $\sigma_{\boldsymbol{r}}$, so a scale-invariant version based on $\tilde{\boldsymbol{r}} = \boldsymbol{r}/\sigma_{\boldsymbol{r}}$ is preferred. A robust estimation for this scale, $\hat{\sigma}_{\boldsymbol{r}}$, is the normalized median absolute deviation (MADN), which is a robust measure of dispersion:

$$\sigma_{\boldsymbol{r}} \approx \hat{\sigma}_{\boldsymbol{r}} \equiv \text{MADN} = \text{MAD}/K, \text{ with MAD} = \text{median}\,|r_i - \text{median}(\boldsymbol{r})| \text{ and } K = \Phi^{-1}(3/4), \tag{8}$$

where Φ is the cumulative distribution function of the standard normal distribution. $M-$estimators may be vulnerable to high-level leverage observations due to unusual design points, but this effect is minimized in our case due to our use of regularly/symmetrically spaced sampling grids.
Even with these approaches, gross outliers can still have a considerable negative effect on the model. Moreover, when the number of observations N is smaller than P, LS produces an underdetermined matrix system. This is why we also wish to benefit from the robustness of the ℓ_1-norm type regression techniques described in the next section.

2.3 Least squares minimization with ℓ_1-regularization

When a function admits a sparse representation, the sparsest representation is obtained by solving this optimization problem:

$$\boldsymbol{u} = \underset{\mathbf{u}\in\mathbb{R}^P}{\text{argmin}} \|\mathbf{u}\|_0 \text{ subject to } \boldsymbol{\Psi}_{\Lambda_p}\mathbf{u} = \mathbf{y}. \tag{9}$$

The ℓ_0-norm of $\boldsymbol{u}$ is just the number of non-zero entries, it is a measure of the sparsity of $\boldsymbol{u}$. This problem, however, is a combinatorial optimization problem: one needs to go through all possible combinations of the columns of $\boldsymbol{\Psi}_{\Lambda_p}$ to find the sparsest solution which is computationally too expensive. One can approximate problem (9) instead by an ℓ_1-optimization problem called *basis pursuit*. This problem is convex and can be solved using linear programming:

$$\boldsymbol{u} = \underset{\mathbf{u}\in\mathbb{R}^P}{\text{argmin}} \|\mathbf{u}\|_1 \text{ subject to the constraint } \|\boldsymbol{y} - \boldsymbol{\psi}_{\Lambda_p}\mathbf{u}\|_2 = 0, \tag{10}$$

where $\|\boldsymbol{u}\|_1 = \sum_{j=1}^{P} |u_j|$. When the observations are noisy, the constraint is too strict and needs to be relaxed. If the magnitude of the noise is bounded: $\|\boldsymbol{e}\|_2 = \epsilon$, then we may write:

$$\boldsymbol{u} = \underset{\mathbf{u}\in\mathbb{R}^P}{\text{argmin}} \|\mathbf{u}\|_1 \text{ subject to } \|\boldsymbol{y} - \boldsymbol{\psi}_{\Lambda_p}\mathbf{u}\|_2 \leq \delta, \tag{11}$$

with $\delta \geq \epsilon$. This problem is sometimes called *basis pursuit denoising*. It is also a convex minimization problem. One can rewrite equation (11) as a corresponding optimization problem in Lagrangian form yielding the so-called LASSO estimate:

$$\boldsymbol{u} = \underset{\mathbf{u}\in\mathbb{R}^P}{\text{argmin}} \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{\psi}_{\Lambda_p}\mathbf{u}\|_2^2 + \lambda\|\mathbf{u}\|_1, \tag{12}$$

where an appropriate $\lambda = \lambda(\boldsymbol{y}, \delta)$ is required. In practice the right value of λ depends on the realizations of the underlying random variables more than the random variables themselves. Therefore the delicate selection of this parameter is often left to cross-validation techniques in order to avoid overfitting. The systems (11) and (12) are equivalent under certain conditions [15] and depending on the formulation one chooses, one of several existing solution techniques can be used to compute $\boldsymbol{u}$ [1, 17, 32, 46]. Several approaches have been recently proposed in order to enhance the efficiency of the representation resulting from solving Equation (11) or (12): – (*a priori* or iteratively) re-weighted ℓ_1-minimization: $\boldsymbol{u} = \text{argmin}_{\mathbf{u}} \|\boldsymbol{W}\,\mathbf{u}\|_1$ where $\boldsymbol{W}$ is diagonal weight matrix, subject to $\|\boldsymbol{y} - \boldsymbol{\psi}_{\Lambda_p}\mathbf{u}\|_2 \leq \delta$ in order to enhance sparsity [33, 53], – better sampling strategies minimizing the mutual coherence of $\boldsymbol{\Psi}_{\Lambda_p}$ [22], – Bayesian compressive sensing [38], or – adaptive basis selection [25].

[4] This class of estimators may be regarded as a generalization of "maximum likelihood" estimation, and hence the capital M designation.

[5] This objective function must satisfy certain properties (non-negativity, symmetry, monotonicity in $|\boldsymbol{r}|$, $\rho(0) = 0$). For ordinary LS regression in the case of error terms that are i.i.d and normally distributed, then $\rho(\boldsymbol{r}) \sim \boldsymbol{r}^2$. For robust regressions, the goal is to minimise some sum of less rapidly increasing function of $\boldsymbol{r}_j$.

In our work, we will solve formulation (12), the Least Absolute Shrinkage and Selection Operator (LASSO), to compute the coefficients. The solution will be computed for a range of values of λ, and using cross-validation, the λ value that is best suited will be retained. The cross-validation used in LASSO is independent from the cross-validation used to determine the weight of each observation. As an alternative, one could use the result from [18] where a value for λ is computed that guarantees, under certain conditions, that the true sparse representation will be recovered.

2.4 Model validation

In the absence of model error $\boldsymbol{e}_M$, *truncation* and *aliasing/sampling* error are the two main potential sources of error in ℓ_2-based approximations. One can further distinguish internal from external aliasing errors [13]. The former exists when the number and position of the samples do not guarantee the numerical discrete orthogonality *within* the chosen expansion basis. In practice, one can check this by verifying if one of these equalities are satisfied: $\boldsymbol{\Psi}_{\Lambda_p}^{\mathsf{T}}\boldsymbol{\Psi}_{\Lambda_p} = \boldsymbol{I}$ or $\mathbb{E}[\psi_i\psi_j] = \delta_{ij}$. For pseudospectral gPC approximations, it is for instance very easy to choose a (sparse) quadrature rule that insures null internal aliasing errors [37]. Inversely, based on a given (sparse) quadrature, we know the (sparse) structure and the order $p_{\max}$ of the polynomial approximation basis we can afford. In this case, the truncation error $\boldsymbol{e_T}$, already defined before, will remain. One way to minimize its contribution is to perform cross-validation of the stochastic approximation in order to identify the optimal approximation space frontier (e.g. $\mathbb{P}_{\Lambda_{p_{\text{opt}}}} \subseteq \mathbb{P}_{\Lambda_{p_{\max}}}$) for the functional of interest. This procedure is also appealing in the presence of model error $\boldsymbol{e}_M$ because cross-validation can reduce the sensitivity to data outliers. This is particularly true for functions that have a smooth noiseless component $\tilde{y}$. In section 3, we will show how we use a leave-one-out cross-validation approach as a first step for the preconditioning strategy of the LS minimization with ℓ_1-regularization.

Resorting to CS techniques in order to exploit potential sparsity of the QoI is interesting because it allows the exploration of a larger approximation space for the same sampling budget. It is therefore a way of reducing the truncation error of the approximation at no cost. In terms of model validation, these techniques, with their built-in property to perform basis selection, also prevent overfitting to some degree. In the LASSO formulation, cf. for instance Eq. (12), there is a *data fidelity* term related to the ℓ_2-norm and a *sparsity* term in the ℓ_1-norm. The LASSO evaluates the coefficients as a trade-off between these two terms thanks to the adjustment of the λ tolerance parameter. The latter may be determined again from cross-validation, e.g. [23, 4]. In this paper, we use a $K = N-$fold cross-validation in the 1D examples and $K = 10$ in the higher dimensional test cases ($K = 10$ is usually a good choice for model selection [5]). There is still one more ingredient that may be added, that is the preconditioning of the data fidelity term. This may be done by assigning some weights or "trust indices" to the samples. Again cross-validation is a handy numerical tool used to evaluate the weight of each sample and this is the second step of our preconditioning strategy.

3 Preconditioning and weight selection

In this section we explain in detail how we derive the observation weights. We aim to assign small weights to observations in which we have a low level of confidence whilst granting a higher weight to observations which we think are reliable. Weighing the observations is a customary technique used in LS regression when one wants to filter out noise:

$$\boldsymbol{u} = \underset{\mathbf{u}\in\mathbb{R}^P}{\operatorname{argmin}} \|\boldsymbol{W}\boldsymbol{y} - \boldsymbol{W}\boldsymbol{\psi}_{\Lambda_p}\mathbf{u}\|_2. \tag{13}$$

The solution to Eq. (13) can be computed as follows:

$$\boldsymbol{u} = \left(\boldsymbol{\Psi}_{\Lambda_p}^{\mathsf{T}}\mathbf{W}\boldsymbol{\Psi}_{\Lambda_p}\right)^{-1}\boldsymbol{\Psi}_{\Lambda_p}^{\mathsf{T}}\mathbf{W}\mathbf{y}, \tag{14}$$

where $\mathbf{W}$ is a diagonal $N \times N$ matrix containing the observation weights. It is also interesting to note that when the sample points are taken as the abscissa of an appropriate quadrature rule and one chooses the diagonal of $\mathbf{W}$ to be composed of the quadrature weights, then formulations (14) and (5) are equivalent.

Appointing weights to the observations can also be done in a compressed sensing framework, the weighted equivalent of Eq. (11) is:

$$\boldsymbol{u} = \underset{\mathbf{u}\in\mathbb{R}^P}{\operatorname{argmin}} \|\mathbf{u}\|_1 \quad \text{subject to} \quad \|\boldsymbol{W}\boldsymbol{y} - \boldsymbol{W}\boldsymbol{\psi}_{\Lambda_p}\mathbf{u}\|_2 \leq \delta. \tag{15}$$

Analogously, one can formulate the weighted equivalent to Eq. (12) as:

$$\boldsymbol{u} = \underset{\mathbf{u}\in\mathbb{R}^P}{\operatorname{argmin}} \frac{1}{2}\|\boldsymbol{W}\boldsymbol{y} - \boldsymbol{W}\boldsymbol{\psi}_{\Lambda_p}\mathbf{u}\|_2^2 + \lambda\|\mathbf{u}\|_1. \tag{16}$$

It is the formulation above that we will be using and the weights will be computed by cross-validation as will be explained in the sections to follow.

3.1 Cross-validation

Leave-one-out cross-validation is a form of K-fold cross-validation with replacement where $K = N$. One constructs the surrogate model, using a method at will, with all but one of the samples. If the i^{th} sample has been left out in the construction of the surrogate model, we shall call the result $\hat{\boldsymbol{y}}_{\Lambda_p}^{(-i)}$ to indicate that this is the approximated surrogate model in Λ_p, computed without taking into account the i^{th} sample. In the framework of compressed sensing, the use of classical cross-validation has been investigated in [47], where results were obtained regarding the number of samples that need to be withheld for the cross-validation process to ensure an accurate representation of the error. The cross-validation method investigated there however is not the same as LOO, so while these results do not directly apply here, heuristics indicate that the LOO error yields a satisfactory estimate for the mean squared error [31]. The LOO error is usually computed as:

$$\varepsilon_{\Lambda_p}^{\mathrm{LOO}} = \frac{1}{N}\sum_{j=1}^{N}\left(y(\boldsymbol{\xi}_j) - \hat{y}_{\Lambda_p}^{(-i)}(\boldsymbol{\xi}_j)\right)^2. \tag{17}$$

For LS solution methods based on a random sampling strategy, Eq. (17) can be more efficiently computed as :

$$\varepsilon_{\Lambda_p}^{\mathrm{LOO}} = \frac{1}{N}\sum_{j=1}^{N}\left(\frac{y_j - \boldsymbol{\Psi}_{\Lambda_p,j}\,\boldsymbol{u}}{1-h_j}\right)^2, \tag{18}$$

where h_j is the j^{th} diagonal term in the matrix $\boldsymbol{\Psi}_{\Lambda_p}\left(\boldsymbol{\Psi}_{\Lambda_p}^{\mathsf{T}}\boldsymbol{\Psi}_{\Lambda_p}\right)^{-1}\boldsymbol{\Psi}_{\Lambda_p}^{\mathsf{T}}$, and $\boldsymbol{\Psi}_{\Lambda_p,j}$ is the j^{th} row of $\boldsymbol{\Psi}_{\Lambda_p}$. When normalized by a variance estimation, this error is called the training error. In this work we will derive this variance from the robust scale estimate introduced in equation (8).

3.2 Quadrature-based leave-one-out error estimation

Because we have intended to work with quadrature rules, we have to develop an accurate and flexible way of computing LOO errors. That is, every time a point is left out from the grid, quadrature weights of the remaining points need to be adjusted in order to insure adequate polynomial integration capability. In the following, we explain in detail how this is done in $d = 1$ and $d = 2$, the generalization to higher dimensions being straightforward.

In $d = 1$ dimension, let us consider a $N-$point: $\Xi_N = \{\xi_1, \ldots, \xi_N\}$ quadrature rule of polynomial accuracy $(N-1)$: $\mathcal{Q}_{N-1}[\cdot]$, and corresponding nodal weights: $\mathcal{W}_N = \{w_1, \ldots, w_N\}$. We now require the $(N-1)-$ point: $\tilde{\Xi}_{N-1}^{(-i)} = \{\xi_1, \ldots, \xi_{i-1}, \xi_{i+1}, \ldots, \xi_N\}_{i\in\{1\ldots N\}}$ reduced quadratures (which will be missing one point relative to the original grid) to be of accuracy $(N-2)$: $\mathcal{Q}_{N-2}^{(-i)}[\cdot]$, i.e. to integrate exactly all univariate polynomials $\mathbb{P}_{\Lambda_{N-2}}$. Let us decompose a member $y \in \mathbb{P}_{\Lambda_{N-2}}$ in a specific basis: i.e. the Lagrange basis L constructed from the discrete nodal values $\tilde{\Xi}_{N-1}^{(-N)}$: i.e. we left out the last point, ξ_N, for simplicity of notation but the result holds for any other dropped point:

$$y(\xi) = \sum_{j=1}^{N-1} y(\xi_j) L_j(\xi), \tag{19}$$

where $L_j(\xi)$ is the Lagrange polynomial associated to ξ_j. Moving to the expectations, we have:

$$\mathbb{E}\left[y(\xi)\right] = \sum_{j=1}^{N-1} y(\xi_j)\mathbb{E}\left[L_j(\xi)\right], \tag{20}$$

We call the new weights $\tilde{\mathcal{W}}_{N-1}^{(-N)} = \{\tilde{w}_1, \ldots \tilde{w}_{N-1}\}$. These new weights should satisfy exact integration of y. It then follows quite naturally from Eq. (20) that these weights should be $\tilde{w}_i = \mathbb{E}\left[L_i(\xi)\right]$, for $i = 1, \ldots, N-1$, which may be evaluated in turn from the full original quadrature:

$$\begin{aligned}\tilde{w}_i &= \sum_{j=1}^{N} w_j L_i(\xi_j) = w_i L_i(\xi_i) + w_N L_i(\xi_N) \quad \left(\text{because } L_i(\xi_j)|_{j=_{j\neq i}1,\ldots,N-1} = 0\right)\\ &= w_i + w_N L_i(\xi_N), \text{ for } i = 1, \ldots, N-1.\end{aligned} \tag{21}$$

The updated weights of the truncated quadrature are made of a summation of the weights from the full quadrature plus the Lagrange polynomial contributions evaluated at the missing node weighted by the original weight of that node. The new weights add up to unity:

$$\sum_{j=1}^{N-1}\tilde{w}_j = \sum_{j=1}^{N-1}\left(w_j + w_N L_j(\xi_N)\right) = \sum_{j=1}^{N-1} w_j + w_N \sum_{j=1}^{N-1} L_j(\xi_N) = 1 - w_N + w_N \times 1 = 1,$$

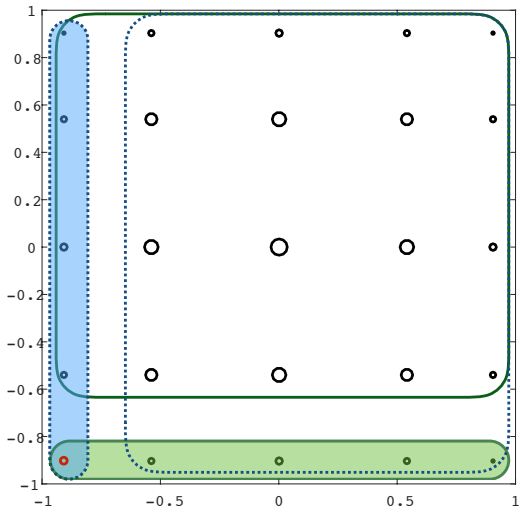

Figure 1: Schematic illustration of quadrature breaking down for leave-one-out error estimation (here for a Gauss-Legendre grid in $d = 2$ dimensions). Quadrature weight magnitudes (before adjustment) are proportional to the circle diameters.

and the truncated quadrature is now valid.
The global LOO error is now evaluated on the full quadrature grid as:

$$\varepsilon_{\Lambda_p}^{\mathrm{LOO}} = \sum_{j=1}^{N} w_j \, r_{\Lambda_p,j}^2, \;\; \text{with} \;\; r_{\Lambda_p,j} = \left(y(\xi_j) - \hat{y}_{\Lambda_p}^{(-i)}(\xi_j) \right), \tag{22}$$

where $\hat{\boldsymbol{y}}_{\Lambda_p}^{(-i)}$ are constructed on the truncated quadratures with adjusted weights $\tilde{\mathcal{W}}_{N-1}^{(-i)}$, for $i \in \{1, \ldots, N\}$.
In higher dimensions, it is common practice to rely on the assumption of independence of the random variables to construct full-grid cubatures[6], that are tensor-products of one-dimensional quadrature rules. In our case, we perform the tensorization between one-dimensional quadratures with different number of points and integration power.
In $d = 2$ dimensions for instance, we build the quadratures by tensorizing the first and the second dimension. Dropping one point along the first direction results in the case of a *truncated* quadrature rule along the first dimension and a full rule along the second dimension; we form quadratures of the type: $\left(\mathcal{Q}_{N_1-2}^{(-i_1)} \otimes \mathcal{Q}_{N_2-1}\right)_{i_1=1\ldots N_1}[\cdot]$, which are exact for any polynomials[7] from $\mathbb{P}_{\Lambda_{N_1-2}} \otimes \mathbb{P}_{\Lambda_{N_2-1}}$, based on a $\tilde{\Xi}_{N_1-1}^{(-i_1)} \times \tilde{\Xi}_{N_2}$ grid of $(N_1-1)N_2$ points and corresponding $\tilde{\mathcal{W}}_{N_1-1} \times \tilde{\mathcal{W}}_{N_2}$ weights. Due to symmetry of the computational grid, we can also build $\left(\mathcal{Q}_{N_1-1} \otimes \mathcal{Q}_{N_2-2}^{(-i_2)}\right)[\cdot]$, for polynomials from $\mathbb{P}_{\Lambda_{N_1-1}} \otimes \mathbb{P}_{\Lambda_{N_2-2}}$, on $\tilde{\Xi}_{N_1} \times \tilde{\Xi}_{N_2-1}^{(-i_2)}$ grid, with weights $\tilde{\mathcal{W}}_{N_1} \times \tilde{\mathcal{W}}_{N_2-1}$.
Figure 1 shows how we proceed to combine error estimation at a particular grid point based on those quadratures. In this example, we are interested by the first point, i.e. $(i_1, i_2) = (1, 1)$. The first grid retained corresponds to the points selected by the dark blue dashed frame, once the left blue column has been dropped from the full lattice. Based on the remaining points, and once the weights have been adjusted, the updated quadrature is put to use to build a surrogate model of the QoI over the full integration domain. This allows a prediction/error estimation at any point from the shaded blue column, such as the lowest left point in red. Note that we can obtain a model error estimation at that point another way: by dropping the points in the bottom green row. Only the points in the green solid frame would then be retained to construct a surrogate model. These two different errors may be combined in several ways. After some testing, we have opted for the arithmetic mean.
The combined residual error at any particular point $r_{\Lambda_p,\boldsymbol{i}}$ is therefore taken as the mean value of the different errors produced by the ensemble of the d surrogate model designs, each built on $N = N_1 \times \ldots (N_{k=1\ldots d} - (d-1)) \ldots \times N_d$ points.

For the computational setup of the quadratures, all partially truncated grids and corresponding adjusted weights combinations can be stored once and for all, for a given grid. Moreover, this step maybe by facilitated by exploiting the natural symmetry of the original multi-dimensional grid. The evaluation of the LOO errors for a given QoI on that grid are then very fast. As a side remark, we have found that estimating LOO errors from truncated Gauss-based quadratures is not significantly more efficient than estimating errors from

[6] We will keep the quadrature nomenclature, no matter the integral dimensions.
[7] In practice, we choose $N_1 = N_2 \equiv \tilde{N}$ and restrain our approximation space to $\mathbb{P}_{\Lambda_{\tilde{N}-2}^{\mathrm{TD}}}$ in order to build the surrogate model.

quadratures with a lower theoretical integration power such as the Clenshaw-Curtis rule. This is because the truncation automatically deteriorates the integration capability from $\mathbb{P}^{\mathrm{TP}}_{\Lambda_{2N-1}}$ to $\mathbb{P}^{\mathrm{TP}}_{\Lambda_{N-2}}$. In the application section, we will be using Clenshaw-Curtis (CC) or Kronrod-Patterson (KP) quadrature rules.

3.3 LOO-weighted preconditioned ℓ_1-minimization approximation

In this section, we review how the different numerical ingredients introduced previously are put together into the general approximation method we propose. There are essentially three main stages, that can be summarized as follows:

1. Selection of a quadrature rule and level, providing a $N-$point grid. Numerical simulations are then performed at these N sampling points and this grid is conserved for the rest of the method.
2. Evaluation of response sample weights that are a measure of confidence in the data obtained and will serve as a preconditioning in the next step.
3. Construction of the cross-validation preconditioned ℓ_1-minimization approximation using a weighted LASSO procedure in order to promote sparsity in a robust way.

The second stage requires more explanations as it is made from of several courses of action. The main idea is to take advantage of cross-validation for the estimation of prediction error in order to guide our model selection and perform robust model assessment. Here the different steps are: 2.i. to rely on the global LOO error of Eq. (22) to determine the most accurate polynomial approximation of the problem response, with the constraint that the aliasing error must be minimized. Then 2.ii. (this step is optional) in order to make the process more robust, not only the optimal approximation but several levels of approximation that are within a certain error threshold are retained. Finally, 2.iii and 2.iv. sample weights are computed as normalized score functions taken at the (averaged) residual error contribution of the retained approximation(s).
More specifically, here are those main steps, revisited in more detail:

2.i. Designation of optimal total degree approximation space $\mathbb{P}_{\Lambda_{p_{\mathrm{opt}}}}$ for pseudospectral gPC representation of the data: $p_{\mathrm{opt}} = \mathrm{argmin}_{p_l \in \mathbb{N}_0^{p_{\max}}} \varepsilon^{\mathrm{LOO}}_{\Lambda_{p_l}}$, will provide the lowest truncation error in the LOO cross-validation criterion with the guarantee of no internal aliasing error. $p_{\max}$ is the highest degree authorized by the quadrature while maintaining no internal aliasing error. The different error estimations $\varepsilon^{\mathrm{LOO}}_{\Lambda_{p_l}}$ are computed from Eq. (22).

2.ii. (this step is optional) Choice of a model cross-validation tolerance parameter $\alpha \geq 1$. Definition of "neighbor" approximation spaces $\mathbb{P}_{\Lambda_{p\in\mathcal{L}}}$ with $\mathcal{L} = \{l \in \{1,\dots,p_{max}\} \,|\, \varepsilon^{\mathrm{LOO}}_{\Lambda_l} \leq \alpha \cdot \varepsilon^{\mathrm{LOO}}_{\Lambda_{p_{\mathrm{opt}}}}\}$ with lower errors than threshold and that will be used in the following.

2.iii. Collect the residual errors at each grid point for the retained surrogate models: $r_{\Lambda_{l\in\mathcal{L}},\, i\in\{1,\dots,N\}}$.

2.iv. Estimation of the (averaged) preconditioning weights as:

$$\mathrm{w}_i = \frac{1}{|\mathcal{L}|} \sum_{l\in\mathcal{L}} \frac{\upsilon(r_{\Lambda_l,i})}{r_{\Lambda_l,i}}, \;\; \forall i = 1,\dots,N, \tag{23}$$

where $|\mathcal{L}|$ is the set cardinality. This averaging is not always necessary (i.e. if $\alpha = 1$ and $l = p_{\mathrm{opt}}$) but sometimes helps in particular when the LOO error function is not clearly convex nor the choice of p_{opt} sharp. It is in some sense reminiscent of the damped version of the re-weighting procedure of Peng *et al.* on p.8 [33].
In this work, Huber, Tukey bisquare and Cauchy score functions have been tested [26] in the numerical applications.

The third stage then consists of the weighted regression and regularization on a space of approximation of total degree larger than the one identified in step 2. i.

The algorithmic complexity and scaling of the computational framework just underlined can be split in different components. The main effort obviously lies in 1. the solution sampling at the grid points. Full cubatures scale exponentially $\mathcal{O}(N^{(d)})$ while sparse cubatures somewhat alleviate the cost $\mathcal{O}\big(N^{-r}(\log N)^{(d-1)(r+1)}\big)$, especially if the solution has high bounded mixed partial derivatives of order r and is isotropic. Then, 2. the determination of response sample weights requires cross-validation evaluations that involve multiple pseudospectral projections and arithmetic averaging. This part is computationally very efficient, even for a large number of dimensions, as long as adjusted weights necessary to the truncated cubatures have been tabulated and stored prior to the computation (cf. discussion at the end of section 3.2). Finally, 3. a regularized weighted regression must be carried out. The computational cost of a ℓ_1 LASSO-type minimization associated to the problem of Eq. (12) is always more expensive than ordinary or weighted LS methods for the same problem. This is due to the "search" for the best parameter λ. Nevertheless, we have noted that our LOO-weighted version sped up the computation. This is due to the preconditioning of the solution. Computational savings differ depending on the problem size and complexity. We have noted savings $5-40\%$ in computational time (savings are more substantial for larger sample points number N). Further improvements may make use of the preconditioning information in order to restrain the search range of λ.
The proposed method will now be demonstrated on several illustrative test problems of different dimensionality, sparsity and complexity.

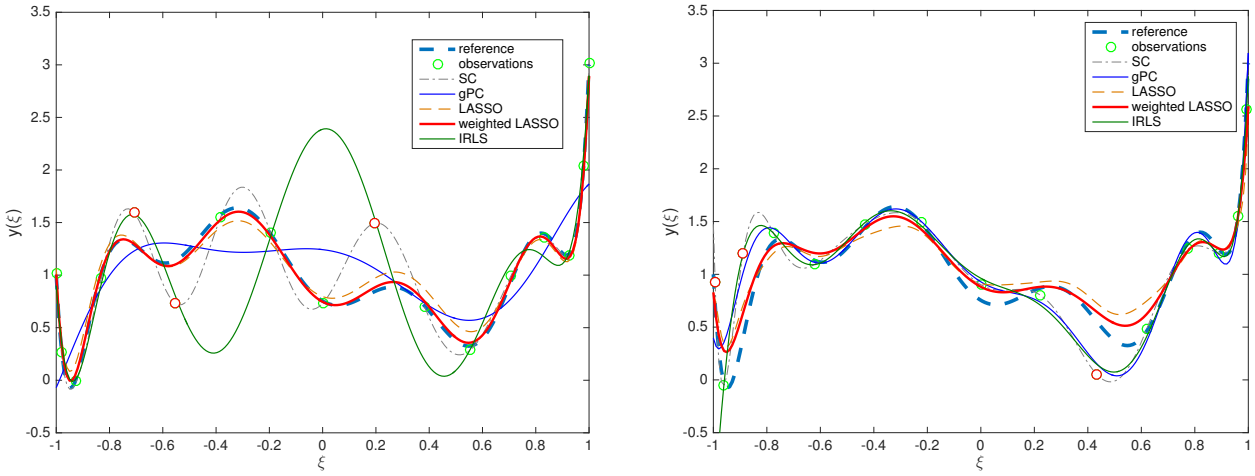

Figure 2: Continuous response surfaces of sparse Legendre polynomials obtained from different approximation methods based on a CC grid of level $l = 5$ (left) and KP $l = 4$ (right). Randomly selected outliers data are identified by red circles. Green circles represent normally distributed data samples subject to background stochastic noise. The SC curve refers the stochastic collocation based on Lagrange interpolation; IRLS is an iterative reweighted least square approximation with the same predictors as the gPC pseudospectral representation. The reference curve is the target noiseless QoI response.

4 Numerical Examples

4.1 Sparse polynomial test functions

The point of these tests is to check the robustness through an analysis of the mechanisms of the proposed method on the approximation of sparse nonlinear polynomial functionals corrupted by few randomly selected data outliers, accounting for deterministic noise. Consider the function $z(\xi) = \mathbb{P}_{10}(\xi) + \mathbb{P}_3(\xi) + \mathbb{P}_0(\xi)$, where $\mathbb{P}_k$ is the k^{th} degree univariate Legendre polynomial, known at some discrete points, in the presence of a stochastic noise component, we have: $y_i = z(\xi_i) + \chi_i$, with $i \in \{1, \ldots, N\}$ and χ_i are centered i.i.d. random variables distributed according to $\mathcal{N}(0, \sigma_\chi)$. These data also contain some outliers that do not match this definition. In practice we have considered σ_χ values about one order of magnitude lower than the variability scale associated to the outliers. The noiseless version of this functional has been previously tested with iterative adaptive polynomial approximations [35]. Here, the random variable $\xi \sim \mathcal{U}_{[-1,1]}$ is *uniformly* distributed. Continuous approximations will be constructed from discrete sampling on regular grids. Without loss of generality, we will be presenting 1. a CC quadrature rule of level $l = 5$ (17 points) and 2. a KP quadrature rule of level $l = 4$ (15 points). Finer grids have also been tested with success.

For case 1., the function we try to approximate by projection is of maximum order 10 which is out reach for the polynomial integration capability of our grid. Standard pseudospectral methods are not able to capture the correct solution in this case, but the function being sparse (only 3 active basis functions are needed), we expect that the ℓ_1-regularization term will help in approaching the right solution. As stated before, we will be using LASSO in order to solve Eq. (12), but our proposed technique for weighting the observations can also be used in combination with other solution methods. The results are presented in Figure 2-(left). The outliers data points are plotted as red open circles, the other points as green open circles. The number of outliers are arbitrarily chosen and affect $N_o = \kappa(\%) \times N$ samples (with $\kappa \approx 18\%$), while $\chi_i \sim \mathcal{N}(0, 4 \cdot 10^{-2})$. The chosen example is tricky as the outliers are placed within the $[\min_{\xi\in[-1,1]} y(\xi), \max_{\xi\in[-1,1]} y(\xi)]$ range. The reference noiseless curve is depicted as a full dashed light blue line. The three solutions that are clearly off the marks are the standard gPC (full blue line), the IRLS (full green line) and the stochastic collocation (thin dotted-dashed gray line) which are also shown for sake of completeness and exhibit too little or too large oscillations. LASSO solution (thin purple dashed line) performs better but not as good as the LOO-weighted LASSO. It is clear that the preconditioned ℓ_1-regularized approximations perform best. These qualitative observations are quantitatively confirmed in Table 1. Table 1 shows the errors in the ℓ_1, ℓ_∞ norm and the R^2 (*goodness of fit*) and the errors in the global statistical moments. Results with stochastic noise-free (green) samples but bearing the same outlying (red) cases are also included.

In Figure 3 some of the internal workings of the proposed technique are exposed. Subplot (a) shows how the cross-validation with LOO technique clearly predicts, despite the noise, that polynomial approximation of total degree $p = 3$ will minimize prediction errors within the range of affordable polynomial orders. This is coherent with the 0^{th}- and 3^{rd}-order components present in the functional. Error estimation solely based on domain integrated gPC residuals are lower as expected but less robust and the optimal polynomial order choice within $\{3, \ldots, 8\}$ is therefore less obvious. Subplot (c) shows the weights assigned to the samples for a Cauchy score function. As expected, levels of confidence are lower for data outliers (represented by red

Grid	Approximation	p^{TD}	μ	σ^2	R^2	ℓ_1	ℓ_∞
$CC_{l=5}$	gPC	8	$4.04{\cdot}10^{-2}$	$3.97{\cdot}10^{-1}$	$2.90{\cdot}10^{-1}$	5.70	1.13
	LASSO	16	$2.82{\cdot}10^{-2}$	$3.25{\cdot}10^{-1}$	$9.74{\cdot}10^{-1}$	1.53	$1.61{\cdot}10^{-1}$
	weighted$_{\alpha=1}$-LASSO	(3) 16	$\mathbf{3.99{\cdot}10^{-4}}$	$\mathbf{1.23{\cdot}10^{-1}}$	$\mathbf{9.96{\cdot}10^{-1}}$	$\mathbf{6.06{\cdot}10^{-1}}$	$1.08{\cdot}10^{-1}$
	weighted$_{\alpha=1.35}$-LASSO	(3) 16	$1.03{\cdot}10^{-2}$	$1.61{\cdot}10^{-1}$	$9.92{\cdot}10^{-1}$	$8.51{\cdot}10^{-1}$	$\mathbf{9.32{\cdot}10^{-2}}$
	IRLS	8	$5.04{\cdot}10^{-2}$	1.53	$5.01{\cdot}10^{-1}$	5.69	1.63
$CC_{l=5}$	gPC	8	$5.08{\cdot}10^{-2}$	$4.05{\cdot}10^{-1}$	$2.93{\cdot}10^{-1}$	5.70	1.12
	LASSO	16	$4.16{\cdot}10^{-2}$	$3.42{\cdot}10^{-1}$	$9.68{\cdot}10^{-1}$	1.64	$1.93{\cdot}10^{-1}$
	weighted$_{\alpha=1}$-LASSO	(3) 16	$\mathbf{1.92{\cdot}10^{-2}}$	$\mathbf{1.61{\cdot}10^{-1}}$	$\mathbf{9.94{\cdot}10^{-1}}$	$\mathbf{7.50{\cdot}10^{-1}}$	$\mathbf{8.53{\cdot}10^{-2}}$
	weighted$_{\alpha=1.35}$-LASSO	(3) 16	$2.47{\cdot}10^{-2}$	$1.84{\cdot}10^{-1}$	$9.92{\cdot}10^{-1}$	$8.88{\cdot}10^{-1}$	$1.04{\cdot}10^{-1}$
	IRLS	8	$9.49{\cdot}10^{-2}$	$7.20{\cdot}10^{-1}$	$5.69{\cdot}10^{-1}$	5.16	1.41

Table 1: In reference to the results of Figure 2-(left): overview of different functional error indicators for a sparse polynomial test case with (top) and without (bottom) stochastic background noise and for different choices of α. The best overall result in bold. $p^{(TD)}$ is the chosen total degree of the polynomial approximation basis; for the weighted-LASSO approach, the value in parenthesis is the optimal order obtained from the original pseudospectral gPC representation resulting in the lowest overall cross-validation LOO error.

Grid	Approximation	p^{TD}	μ	σ^2	R^2	ℓ_1	ℓ_∞
$KP_{l=4}$	gPC	10	$\mathbf{1.48{\cdot}10^{-3}}$	$\mathbf{2.10{\cdot}10^{-1}}$	$8.47{\cdot}10^{-1}$	2.58	$6.18{\cdot}10^{-1}$
	LASSO	20	$5.62{\cdot}10^{-2}$	$5.57{\cdot}10^{-1}$	$7.43{\cdot}10^{-1}$	2.39	$5.25{\cdot}10^{-1}$
	weighted-LASSO	(10) 20	$4.48{\cdot}10^{-2}$	$3.58{\cdot}10^{-1}$	$\mathbf{9.17{\cdot}10^{-1}}$	**1.56**	$\mathbf{3.07{\cdot}10^{-1}}$
	IRLS	10	$1.92{\cdot}10^{-2}$	$4.27{\cdot}10^{-1}$	$5.89{\cdot}10^{-1}$	3.82	1.80

Table 2: Same caption as in Table 1, but in reference to the results of Figure 2-(right).

circles). They are also low for the boundary samples that are negatively affected due to their distance to the low-order approximation. Last two subplots show the subtle differences in the mean square errors (MSE) distribution vs. λ for the LASSO (d) and the weighted-LASSO (e). Very low values of λ, to the right of these plots, lead to approximations dominated by the first term of Eq. (16). Despite the preconditioning, the ℓ_2 minimization alone produces larger errors with large error bars. Once the optimal λ selected, very low MSE errors are obtained and coefficients amplitude in subplot (b) shows that the three leading modes of the functional, including the tenth-order, are almost perfectly captured, despite some weak spurious energy in the 5^{th} and 12^{th} modes.

The next test case has a similar setup but a larger noise, e.g. $\chi_i \sim \mathcal{N}(0, 7 \cdot 10^{-2})$ and consequently more severe data outliers, on a different sampling grid. For case 2., the KP quadrature/grid combination has a higher integration capability than the previous grid. This time, results presented in Figure 2-(right) are not visually as impressive, but weighted-LASSO still performs best in most of the error norms, cf. Table 2.

Additional results collected in Figure 4 better point to some of the differences with the previous case. This time, the cross-validation is able to predict that polynomial of order 10 is also crucial to the approximation. Data outliers are then endowed with low confidence but a few other data samples are misleadingly granted with low weights as well (c). The LASSO error distribution plots (d-e) show that the error levels remain low even for very small values of λ. In this case, the ℓ_1-regularization term does not contribute significantly in terms of the accuracy improvement. However, the preconditioning still helps the LASSO algorithm in better finding the optimal λ value. Looking at the emerging modal coefficients in (b), we notice again that despite its better results, weighted-LASSO is not as sparse as the standard LASSO approximation.
Other one-dimensional tests were pursued in the same spirit, for non-sparse non-polynomial functions. For instance, results for a noisy data set obtained from $z(\xi) = (-3\xi^5 + \xi^2 + \xi) \times \tanh(\xi)$ and corrupted by four data outliers (results not presented here) confirmed the performance advantage of weighted-LASSO with respect to LASSO and standard gPC.

4.2 Higher-dimensional non-polynomial test function

Now, we consider a higher-dimensional test function that is not necessarily compressible so that we do not favor ℓ_1-type regression method over robust iterative weighted least-square approximations. We assess the continuous approximation of an algebraic noisy version of the Genz corner-peak function, known at some discrete locations, which provides a flexible test for the proposed method:

$$y_i = \left(1 + \sum_{k=1}^{d} c_k \xi_i^{(k)}\right)^{-(d+1)} + \chi_i, \text{ with } i = 1, \ldots, N, \tag{24}$$

and χ_i are centered i.i.d. random variables distributed according to $\mathcal{N}(0, \sigma_\chi)$ and $\boldsymbol{\xi} \equiv (\xi^{(1)}, \ldots, \xi^{(d)}) \sim \mathcal{U}_{[0,1]^d}$. Specifically, the coefficients c_k can be used to control the effective dimensionality and the compressibility of this function. In the following, we first test the $d = 3-$dimensional version using the anisotropic coefficients $c_k = 1/k^2$ defined in [25]. The function is computed on a 7^3 KP grid but similar tests have been

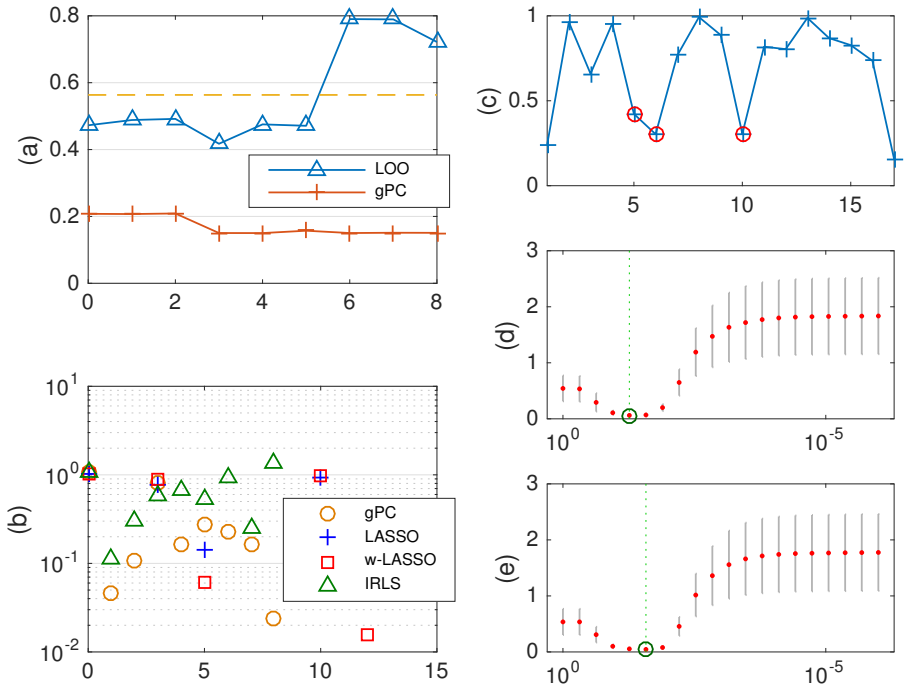

Figure 3: In reference to the results of Figure 2-(left): (a) overview of the model errors vs. polynomial total degree p, (b) polynomial coefficients magnitude u_j^2 vs. p, (c) sample weights and finally model cross-validated mean square errors vs. λ for LASSO (d) and for weighted-LASSO (e).

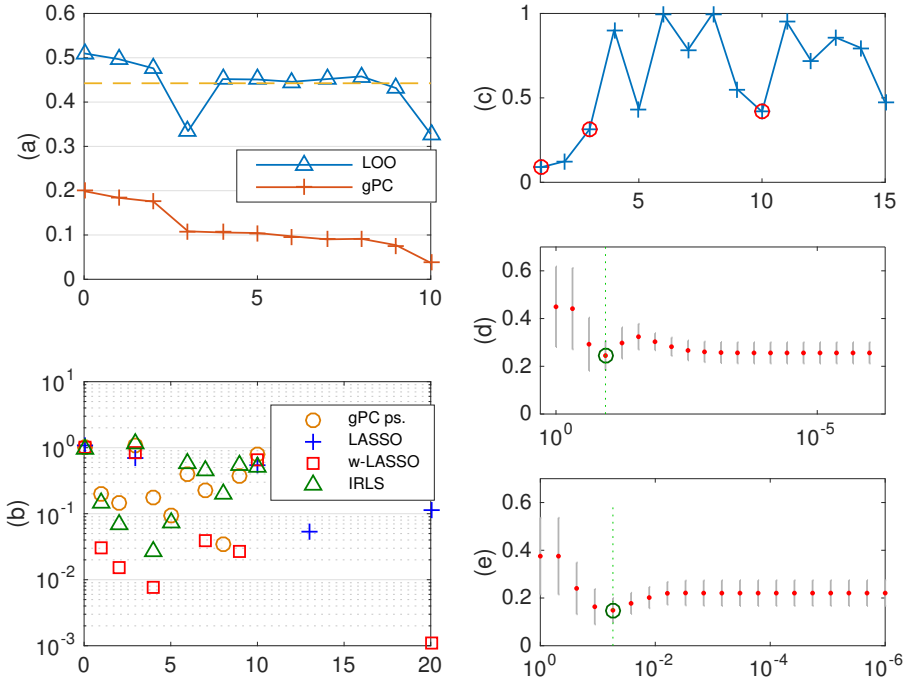

Figure 4: Same caption as in Figure 3, but in reference to the results of Figure 2-(right).

Grid	Approximation	p^{TD}	$\overline{\mu}$	$\overline{\sigma^2}$	$\overline{R^2}$	$\overline{\ell_1}$	$\overline{\ell_\infty}$
$\mathrm{KP}_{l=3}$	gPC	5	$1.46e{-}02$	$2.07e{-}01$	$8.33e{-}01$	$1.75e{+}01$	$3.10e{-}01$
	LASSO	9	$1.14e{-}02$	$1.52e{-}01$	$9.60e{-}01$	$7.24e{+}00$	$9.49e{-}02$
	weighted-LASSO	(1.2) 9	$6.82e{-}03$	$6.55e{-}02$	$9.89e{-}01$	$\mathbf{3.88e{+}00}$	$5.38e{-}02$
	IRLS	5	$\mathbf{3.67e{-}03}$	$\mathbf{3.21e{-}02}$	$\mathbf{9.91e{-}01}$	$4.05e{+}00$	$\mathbf{5.05e{-}02}$

Table 3: Similar caption as Table 1 for $d = 3$ dimensions Genz corner-peak functional. But this time, all errors are averaged as the test cases have been repeated 500 times for different initial conditions.

Grid	Approximation	p^{TD}	μ	σ^2	R^2	ℓ_1	ℓ_∞
$\mathrm{KP}_{l=3}$	gPC	5	$2.70{\cdot}10^{-3}$	$2.88{\cdot}10^{-2}$	$9.71{\cdot}10^{-1}$	$3.71{\cdot}10^{2}$	$2.70{\cdot}10^{-1}$
	LASSO	9	$3.38{\cdot}10^{-3}$	$1.93{\cdot}10^{-2}$	$9.98{\cdot}10^{-1}$	$8.29{\cdot}10^{1}$	$4.44{\cdot}10^{-2}$
	weighted-LASSO	(2) 9	$\mathbf{1.28{\cdot}10^{-3}}$	$\mathbf{5.20{\cdot}10^{-3}}$	$\mathbf{9.99{\cdot}10^{-1}}$	$\mathbf{5.37{\cdot}10^{1}}$	$2.79{\cdot}10^{-2}$
	IRLS	5	$3.35{\cdot}10^{-3}$	$1.04{\cdot}10^{-2}$	$9.99{\cdot}10^{-1}$	$6.06{\cdot}10^{1}$	$\mathbf{1.91{\cdot}10^{-2}}$

Table 4: Same caption as Table 1 for $d = 5$ dimensions Genz corner-peak test case.

performed for finer grids as well as for grids of different nature without affecting the overall conclusion. The outlying cases amount to $N_o = \kappa(\%) \times N$ randomly selected samples in the domain. In practice, the outlier locations are randomly distributed in the computational domain with a uniform distribution. For these high-dimensional cases, their magnitude is automatically drawn from either: - a non-normal distribution or - a normal distribution with a standard deviation of one order of magnitude larger than σ_χ. The latter definition has been used for the results presented next. Moreover, the procedure has been repeated 500 times (i.e. with different initial conditions both for outlier locations and magnitudes and for the stochastic noise). Statistical results are presented in Figure 5. They show that the standard gPC approximation is not robust. LASSO improves the error statistics, in particular in the ℓ_1 and ℓ_∞ norms. Mean values extracted from these distributions and reported in Table 3 confirm these findings.
One-shot LOO-weighted LASSO improves the results even further, coming close to the IRLS iterative scheme. A single test following a similar setup is carried out for $d = 5$ dimensions. We recall that this dimensional limitation closely connected to memory requirement is inherited from the scaling of full cubature sampling grid, but would be alleviated for a sparse cubature. The error results summarized in Table 4 demonstrate again that LOO-weighted LASSO and IRLS are the best two contenders for robustness and that our approach performs well compared to the iterative scheme.

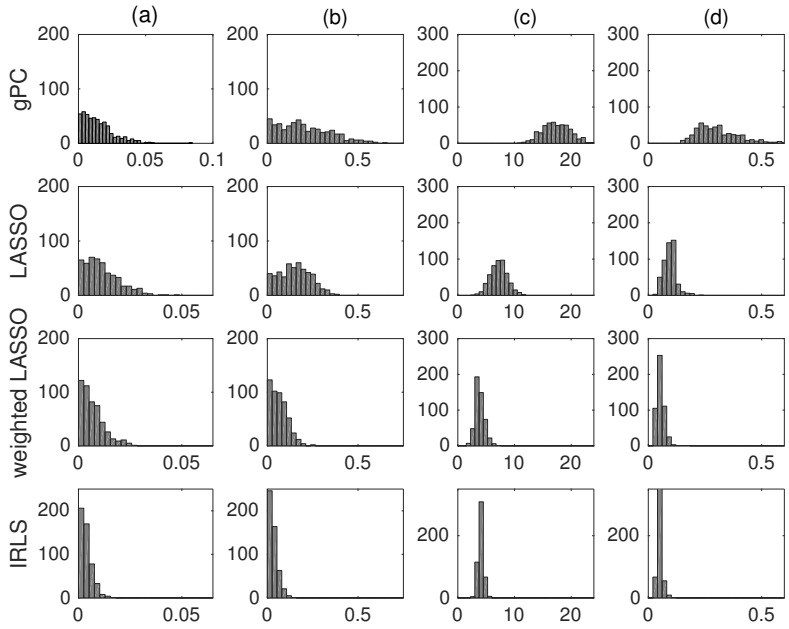

Figure 5: Comparison of different approximation errors: i.e. μ (a), σ^2 (b), ℓ_1 (c) and ℓ_∞ (d), of a three-dimensional non-polynomial noisy Genz corner-peak function obtained from different methods relative to the noise-free reference solution. All approximations are based on 7^3 KP sampling grid. Data outliers affect $\kappa = 15\%$ of the total number of samples and $\sigma_\chi = 0.03$. The test is repeated 500 times for different random initial conditions and stochastic noise.

4.3 2D compressible, inviscid flow over an inclined NACA0012 airfoil

The study of compressible flows around a NACA0012 airfoil are considered in this example. The functional of interest is the stagnation pressure P_a integrated along the airfoil profile Γ :

$$P_a = \frac{1}{L(\Gamma)p_{a_\infty}} \int_\Gamma p_a \mathrm{d}\Gamma, \tag{25}$$

where $p_a = p\left(1 + \frac{\gamma-1}{2}M_\infty^2\right)^{\frac{\gamma}{\gamma-1}}$ (respectively p_{a_∞}) is the (upstream) stagnation pressure, γ is the specific heat ratio, here fixed at 1.4 and the free-stream Mach number $M_\infty = 0.5$, $L(\Gamma)$ denotes the length of the airfoil. We have considered for this analysis one uniformly distributed uncertain parameter: the angle of attack AoA $\equiv \xi \sim \mathcal{U}_{[0;8^o]}$. It is well known that for subcritical Euler flows at zero or moderate angle of incidence, the stagnation pressure should be exactly equal to unity [34]. In practice, even for low Mach numbers this exact value is unreachable due to numerical (e.g. discretization) errors. Moreover, for larger angles of incidence (i.e. AoA $\gtrapprox 6$), the loss of symmetry is such that the flow switches from the subsonic to a transonic regime, with a shock appearing on the upper surface close to the leading edge. Figure 6 illustrates this phenomenon with three snapshots of the density field for AoA $= 0$ (left image), AoA $= 6$ (middle image) and AoA $= 8$ (right image), respectively. The rise of this shock noticeably modifies the flow features and negatively affects accuracy of the prediction if no adjustment is made to the model in order to account for it. It especially impacts the discretization error if the retained mesh is too coarse and not adapted at the shock location. In this case the model error magnitude will depend on the value of the AoA in quite an unpredictable manner as can be seen in the stagnation pressure response pictured in Figure 7 (red circles). The response was obtained for each AoA from the same coarse mesh: with about five thousand mesh cells

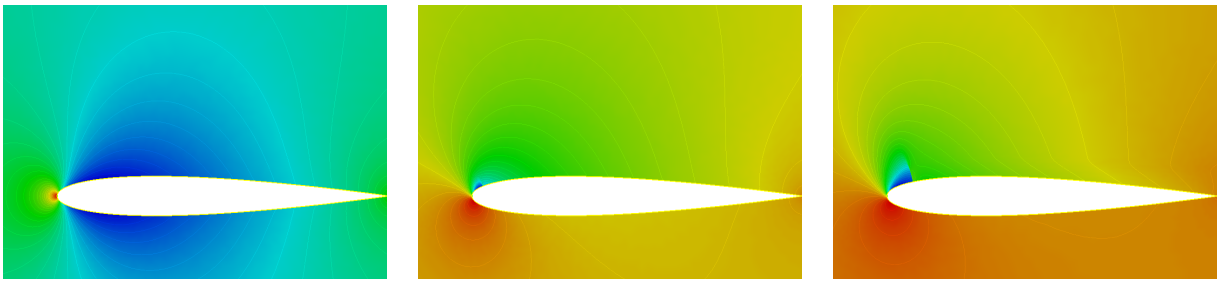

Figure 6: NACA0012: density field closeup for different angles of attack: AoA $= 0$ (left), AoA ≈ 6 (middle) and AoA $= 8$ (right). Note the presence of small shocks close to the leading edge at large angles of attack. Computational meshes are not displayed but have been adapted and refined to capture all relevant flow features.

regularly distributed around the airfoil and referred as the uniform $5K$ mesh. We notice strong oscillations in the transcritical region for AoA larger than about six degrees. Consequently, high-order pseudospectral gPC approximation (dotted blue curve) is corrupted with errors as expected. Cross-validation preconditioned regularized approximation (solid red curve) does much better at filtering out small spurious fluctuations in the left region where P_a is not dependent on the AoA, as well as controlling and erasing large unphysical P_a oscillations on the right hand side of the domain.
Interestingly, for this problem, it is possible and still affordable to produce results that are almost model error-free. By refining the mesh to 38000 mesh cells ($38K$), the discretization error is drastically reduced. These refined meshes are adapted to *each* AoA scenario in order to capture the critical physics (e.g. shocks). Computations on the refined and adapted meshes are represented by the green stars. We observe a flat zone corresponding to low angles of attack where the stagnation pressure is very close to, but lower than unity (due to still present numerical diffusion), followed by a sharp almost linear decrease for larger angles of attack. In this case, our method does not alter the data and produces a smooth response (solid brown curve) while perfectly maintaining the right slope.

If we now consider that the Mach number is also uncertain: for instance, $\mathrm{M}_\infty \sim \mathcal{U}_{[0.3,0.5]}$, the stagnation pressure value departs from unity at a critical angle that depends on the Mach number; this angle being larger for lower Mach numbers. This induces a narrow region with a steep slope that is difficult to capture accurately by standard projection techniques and induces spurious oscillations, cf. Figures 8 and 9. Again the proposed method increases the regularity of the surrogate where it is needed, with no *a priori* information nor significant computational overhead, while capturing relevant local sharp features even on the coarser mesh. We notice in particular that the surface goes through the data samples much better in the transcritical region.

5 Conclusions

The main contribution of this paper was to propose a non-iterative robust numerical method for the uncertainty quantification of reasonably compressible multivariate stochastic solutions. The goal was to make the

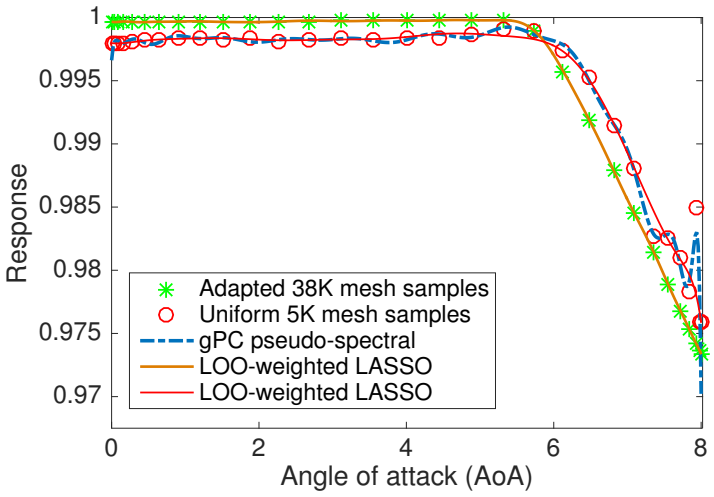

Figure 7: NACA0012: averaged stagnation pressure P_a response surfaces vs. AoA, obtained from different approximation methods based on a level $l = 5$ Kronrod-Patterson data sampling. Two classes of discretization meshes of the Euler flow are investigated: – uniform coarse mesh (red circles) vs. – fine mesh adapted to each flow incidence (green stars).

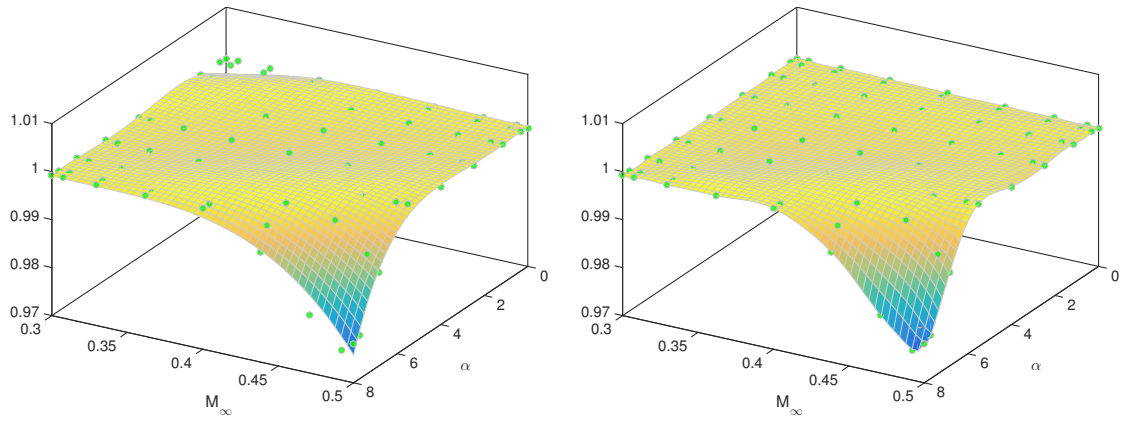

Figure 8: NACA0012: averaged stagnation pressure P_a response surfaces vs. AoA and M_∞, based on a 9^2 Clenshaw-Curtis data sampling: – pseudospectral gPC expansion with $p = 4$ (left) and – LOO-weighted LASSO (right). Each deterministic CFD simulation is performed on a non-adapted mesh with ~ 7000 cells.

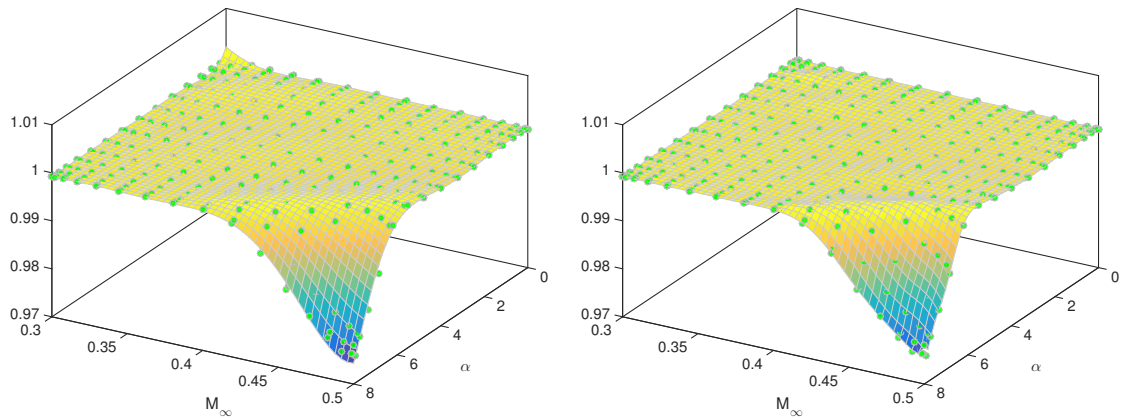

Figure 9: Same caption as the previous figure but based on a 17^2 Clenshaw-Curtis data sampling: – pseudospectral gPC expansion with $p = 8$ (left) and – LOO-weighted LASSO (right).

approximation capable of dampening the effect of outlying data to that do not fit the assumption of small additive stochastic noise represented by centered i.i.d. (normal) random variables with uniformly bounded variance; in particular, noise which does not fall under the regularity assumption of the stochastic trunca-

tion error but pertains to a more complete error model. The method required a preconditioning prior to a dimension reduction of the solution, i.e.: 1. a ℓ_2-based cross-validation of a generalized Polynomial Chaos approximation of the response; this allowed a first model selection and the computation of (preconditioning) weights (i.e. confidence measures) associated to the samples, followed by 2. a preconditioned least-squares polynomial approximation with regularization using the weighted Least Absolute Shrinkage and Selection Operator. For the first step, observation weights were computed from sample contributions to the cross-validation leave-one-out error of the selected surrogate model. For the second step, other algorithms may be used to solve the optimization problem resulting from the ℓ_1-regularization. Numerical test cases treated in this paper have proved the numerical method to be more effective in automatically canceling out or reducing the influence of data outliers than standard compressed sensing techniques and of comparable efficiency to iterative robust regression techniques.
A particularity of this work was to make use of quadrature rules/grids as opposed to random sampling. This zero-variability sampling brings reliability to the recovery procedure but is better suited for low to moderate dimensional problems (with possibly high-order representation). However, the approach remains general and could be applied to higher dimensions by using random sampling or quadrature subsampling schemes taking advantage of recent advances in terms of polynomial recovery optimization. In this case, the use of other cross-validation techniques with potentially lower variance error estimations is also conceivable.
Potential perspectives for future work involve: – a different way of evaluating the preconditioning weights that would be even less sensitive to data outliers. With this aim, one may argue that non-weighted LASSO–type algorithms could be deployed upfront as we have noticed they often performed better in terms of robustness than standard ℓ_2-projections. Once combined with the second step above, this approach could then be generalized in the form of an adaptive formulation where the weights would be iteratively refined in conjunction with the surrogate model level of complexity. In this case, it would be reminiscent of an iteratively reweighted least squares technique; or – introduce a (re)weighted norm in the ℓ_1-minimization which is known to produce better compressive performance. This information could be provided by the spectrum of the low-order model, selected and validated in the initial step.

Acknowledgements

The authors are thankful to Dr. O. Le Maître and Dr. L. Mathelin of LIMSI-CNRS, Orsay, France, for valuable discussions related to some of the aspects of this work.

References

[1] Stephen Becker, Jérôme Bobin, and Emmanuel J Candès. NESTA: a fast and accurate first-order method for sparse recovery. *SIAM Journal on Imaging Sciences*, 4(1):1–39, 2011.

[2] Marc Berveiller, Bruno Sudret, and Maurice Lemaire. Stochastic finite element: a non intrusive approach by regression. *European Journal of Computational Mechanics/Revue Européenne de Mécanique Numérique*, 15(1-3):81–92, 2006.

[3] Hester Bijl, Didier Lucor, Siddharta Mishra, and Christoph Schwab, editors. *Uncertainty quantification in Computational Fluid Dynamics*, volume 92 of *Lecture Notes in Computational Science and Engineering*. Springer, 2013.

[4] G. Blatman and B. Sudret. Adaptive sparse polynomial chaos expansion based on least angle regression. *Journal of Comput. Physics*, 230-6:2345–2367, 2011.

[5] Leo Breiman and Philip Spector. Submodel selection and evaluation in regression. The X-random case. *International statistical review/revue internationale de Statistique*, pages 291–319, 1992.

[6] Robert H Cameron and William T Martin. The orthogonal development of non-linear functionals in series of Fourier-Hermite functionals. *Annals of Mathematics*, pages 385–392, 1947.

[7] E. J. Candès, J. K. Romberg, and T. Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *Information Theory, IEEE Transaction*, 52-2:489–509, 2004.

[8] E. J Candès, J. K. Romberg, and T. Tao. Stable signal recovery from incomplete and inaccurate measurements. *Communications on pure and applied mathematics*, 59(8):1207–1223, 2006.

[9] E. J. J Candès and T. Tao. Decoding by linear programming. *Information Theory, IEEE Transactions on*, 51(12):4203–4215, 2005.

[10] A. Chkifa, A. Cohen, G. Migliorati, F. Nobile, and R. Tempone. Discrete least squares polynomial approximation with random evaluations – application to parametric and stochastic elliptic pdes. *ESAIM: M2AN*, 49(3):815–837, 2015.

[11] Seung-Kyum Choi, Ramana V Grandhi, Robert A Canfield, and Chris L Pettit. Polynomial Chaos expansion with Latin Hypercube Sampling for estimating response variability. *AIAA journal*, 42(6):1191–1198, 2004.

[12] Albert Cohen, Mark A Davenport, and Dany Leviatan. On the stability and accuracy of least squares approximations. *Foundations of computational mathematics*, 13(5):819–834, 2013.

[13] Patrick R Conrad and Youssef M Marzouk. Adaptive Smolyak pseudospectral approximations. *SIAM Journal on Scientific Computing*, 35(6):A2643–A2670, 2013.

[14] David L Donoho. Compressed sensing. *Information Theory, IEEE Transactions on*, 52(4):1289–1306, 2006.

[15] David L Donoho, Michael Elad, and Vladimir N Temlyakov. Stable recovery of sparse overcomplete representations in the presence of noise. *Information Theory, IEEE Transactions on*, 52(1):6–18, 2006.

[16] Alireza Doostan and Houman Owhadi. A non-adapted sparse approximation of PDEs with stochastic inputs. *Journal of Computational Physics*, 230(8):3015–3034, 2011.

[17] Bradley Efron, Trevor Hastie, Iain Johnstone, and Robert Tibshirani. Least angle regression. *The Annals of statistics*, 32(2):407–499, 2004.

[18] Jean Jacques Fuchs. Recovery of exact sparse representations in the presence of bounded noise. *Information Theory, IEEE Transactions on*, 51(10):3601–3608, 2005.

[19] Roger G Ghanem and Pol D Spanos. *Stochastic finite elements: a spectral approach*. Courier Corporation, 2003.

[20] P. J. Green. Iteratively reweighted least squares for maximum likelihood estimation, and some robust and resistant alternatives. *Journal of the Royal Statistical Society. Series B (Methodological)*, 46(2):pp. 149–192, 1984.

[21] Jerrad Hampton and Alireza Doostan. Coherence motivated sampling and convergence analysis of least squares Polynomial Chaos regression. *Computer Methods in Applied Mechanics and Engineering*, 290:73–97, 2015.

[22] Jerrad Hampton and Alireza Doostan. Compressive sampling of Polynomial Chaos expansions: convergence analysis and sampling strategies. *Journal of Computational Physics*, 280:363–386, 2015.

[23] T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning*. Springer series in statistics. Springer, 2nd edition, 2009.

[24] P. J. Huber and E. M. Ronchetti. *Robust statistics*. Probability and statistics. Wiley, 2nd edition, 2009.

[25] John D Jakeman, Michael S Eldred, and Khachik Sargsyan. Enhancing ℓ_1-minimization estimates of polynomial chaos expansions using basis selection. *Journal of Computational Physics*, 289:18–34, 2015.

[26] Raymond J. Carroll James O. Street and David Ruppert. A note on computing robust regression estimates via iteratively reweighted least squares. *The American Statistician*, 42(2):pp. 152–154, 1988.

[27] Olivier P Le Maître and Omar M Knio. *Spectral Methods for Uncertainty Quantification*. Springer, 2010.

[28] L. Mathelin and K. A. Gallivan. A compressed sensing approach for partial differential equations with random input data. *Communications in Computational Physics*, 12(4):919–954, 2012.

[29] Giovanni Migliorati, Fabio Nobile, and Raúl Tempone. Convergence estimates in probability and in expectation for discrete least squares with noisy evaluations at random points. *Journal of Multivariate Analysis*, 142:167 – 182, 2015.

[30] Giovanni Migliorati, Fabio Nobile, Erik von Schwerin, and Raúl Tempone. Analysis of discrete ℓ_2 projection on polynomial spaces with random evaluations. *Foundations of Computational Mathematics*, 14(3):419–456, 2014.

[31] Annette M Molinaro, Richard Simon, and Ruth M Pfeiffer. Prediction error estimation: a comparison of resampling methods. *Bioinformatics*, 21(15):3301–3307, 2005.

[32] Deanna Needell and Joel A Tropp. CoSaMP: Iterative signal recovery from incomplete and inaccurate samples. *Applied and Computational Harmonic Analysis*, 26(3):301–321, 2009.

[33] Ji Peng, Jerrad Hampton, and Alireza Doostan. A weighted ℓ_1-minimization approach for sparse polynomial chaos expansions. *Journal of Computational Physics*, 267:92–111, 2014.

[34] J. Peter, M. Nguyen-Dinh, and P. Trontin. Goal oriented mesh adaptation using total derivative of aerodynamic functions with respect to mesh coordinates – With applications to Euler flows. *Computers & Fluids*, 66:194 – 214, 2012.

[35] G. Poëtte, A. Birolleau, and D. Lucor. Iterative Polynomial Approximation Adapting to Arbitrary Probability Distribution. *SIAM J. Numerical Analysis*, 53(3):1559–1584, 2015.

[36] Holger Rauhut and Rachel Ward. Sparse Legendre expansions via ℓ_1-minimization. *Journal of approximation theory*, 164(5):517–533, 2012.

[37] A. Resmini, J. Peter, and D. Lucor. Sparse grids-based stochastic approximations with applications to aerodynamics sensitivity analysis. *Int. J. Numer. Meth. Engng*, 10.1002/nme.5005, 2015.

[38] Khachik Sargsyan, Cosmin Safta, Habib N Najm, Bert J Debusschere, Daniel Ricciuto, and Peter Thornton. Dimensionality reduction for complex models via Bayesian compressive sensing. *International Journal for Uncertainty Quantification*, 4(1), 2014.

[39] J. Shen and L.-L. Wang. Sparse spectral approximations of high-dimensional problems based on hyperbolic cross. *SIAM Journal on Numerical Analysis*, 48(3):1087–1109, 2010.

[40] Alex J Smola and Bernhard Schölkopf. A tutorial on support vector regression. *Statistics and computing*, 14(3):199–222, 2004.

[41] Gary Tang and Gianluca Iaccarino. Subsampled Gauss quadrature nodes for estimating Polynomial Chaos expansions. *SIAM/ASA Journal on Uncertainty Quantification*, 2(1):423–443, 2014.

[42] Tao Tang and Tao Zhou. On discrete least-squares projection in unbounded domain with random evaluations and its application to parametric uncertainty quantification. *SIAM Journal on Scientific Computing*, 36(5):A2272–A2295, 2014.

[43] M. Tatang, W. Pan, R. Prinn, and G. McRae. An efficient method for parametric uncertainty analysis of numerical geophysical models. *Journal of Geophysical Research*, 102:21925–21932, 1997.

[44] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society*, 58-1:267–288, 1996.

[45] Joel Tropp. Just relax: Convex programming methods for identifying sparse signals in noise. *Information Theory, IEEE Transactions on*, 52(3):1030–1051, 2006.

[46] Ewout Van Den Berg and Michael P Friedlander. Probing the Pareto frontier for basis pursuit solutions. *SIAM Journal on Scientific Computing*, 31(2):890–912, 2008.

[47] Rachel Ward. Compressed sensing with cross-validation. *Information Theory, IEEE Transactions on*, 55(12):5773–5782, 2009.

[48] Norbert Wiener. The Homogeneous Chaos. *American Journal of Mathematics*, pages 897–936, 1938.

[49] Christopher KI Williams and Carl Edward Rasmussen. Gaussian processes for machine learning. *the MIT Press*, 2(3):4, 2006.

[50] Dongbin Xiu and Jan S. Hesthaven. High-order collocation methods for differential equations with random inputs. *J. Sci. Comput.*, 27(3):1118–1139, 2005.

[51] Dongbin Xiu and George Em Karniadakis. The Wiener-Askey polynomial chaos for stochastic differential equations. *SIAM journal on scientific computing*, 24(2):619–644, 2002.

[52] Liang Yan, Ling Guo, and Dongbin Xiu. Stochastic collocation algorithms using ℓ_1-minimization. *International Journal for Uncertainty Quantification*, 2(3), 2012.

[53] Xiu Yang and George Em Karniadakis. Reweighted ℓ_1 minimization method for stochastic elliptic differential equations. *Journal of Computational Physics*, 248:87–108, 2013.

[54] T. Zhou, A. Narayan, and D. Xiu. Weighted discrete least-squares polynomial approximation using randomized quadratures. *J. Comp. Phys.*, 298:787–800, 2015.

[55] Tianhe Zhou and Danfu Han. A weighted least squares method for scattered data fitting. *Journal of Computational and Applied Mathematics*, 217(1):56–63, 2008.