

Kernel-based Active Subspaces with application to CFD problems using Discontinuous Galerkin method

Francesco Romor^{*}, Marco Tezzele[†], Andrea Lario[‡] and Gianluigi Rozza[§]

Mathematics Area, mathLab, SISSA, via Bonomea 265, I-34136 Trieste, Italy

August 3, 2023

Abstract

Nonlinear extensions to the active subspaces method have brought remarkable results for dimension reduction in the parameter space and response surface design. We further develop a kernel-based nonlinear method. In particular we introduce it in a broader mathematical framework that contemplates also the reduction in parameter space of multivariate objective functions. The implementation is thoroughly discussed and tested on more challenging benchmarks than the ones already present in the literature, for which dimension reduction with active subspaces produces already good results. Finally, we show a whole pipeline for the design of response surfaces with the new methodology in the context of a parametric CFD application solved with the Discontinuous Galerkin method.

Contents

1	Introduction	2
2	Active Subspaces for parameter space reduction	3
2.1	Response surfaces	6
3	Kernel-based Active Subspaces extension	7
3.1	Choice of the Feature Map	9
3.2	Random Fourier Features	11
4	Benchmark test problems	12
4.1	Radial symmetric functions	13
4.2	SEIR model for the spread of Ebola	15
4.3	Elliptic Partial Differential Equation with random coefficients	15
5	A CFD parametric application of KAS solved with the DG method	17
5.1	Domain and mesh description	19
5.2	Parameter space description	19
5.3	Governing equations	19
5.4	Numerical results	21
6	Conclusions and perspectives	24
A	Appendix - Proof details	25

^{*}francesco.romor@sissa.it

[†]marco.tezzele@sissa.it

[‡]andrea.lario@sissa.it

[§]gianluigi.rozza@sissa.it

1 Introduction

Nowadays, in many industrial settings the simulation of complex systems requires a huge amount of computational power. Problems involving high-fidelity simulations are usually large-scale, moreover the number of solutions required increases with the number of parameters. In this context, we mention optimization tasks, inverse problems, optimal control problems, and uncertainty quantification; they all suffer from the curse of dimensionality, that is, in this case, the computational time grows exponentially with the dimension of the input parameter space. Data-driven reduced order methods (ROM) [11, 52, 54] have been developed to deal with such costly outer loop applications for parametric PDEs, but the limit for high dimensional parameter spaces remains.

One approach to alleviate the curse of dimensionality is to identify and exploit some notion of low-dimensional structure of the model or objective function that maps the inputs to the outputs of interest. A possible linear input coordinate transformation technique is the Sliced Inverse Regression (SIR) [34] approach and its extensions [16, 35, 74]. Sharing some characteristics with SIR, there is the active subspaces (AS) property¹ [53, 12, 13, 75] which, in the last years, has emerged as a powerful linear data-driven technique to construct ridge approximations using gradients of the model function. AS has been successfully applied to quantify uncertainty in the numerical simulation of the HyShot II scramjet [15], and for sensitivity analysis of an integrated hydrologic model [29]. Reduction in parameter space has been coupled with model order reduction techniques [27, 46, 51] to enable more complex numerical studies without increasing the computational load. We mention the use of AS in cardiovascular applications with POD-Galerkin [60], in nonlinear structural analysis [25], in nautical and naval engineering [66, 62, 63, 61], coupled with POD with interpolation for structural and computational fluid dynamic (CFD) analysis [19, 65], and with Dynamic Mode Decomposition in [64]. Applications in automotive engineering within a multi-fidelity setting can be found in [48], for turbomachinery see [57], while for results in chemistry see [30, 70]. Advances in efficient global design optimization with surrogate modeling are presented in [38, 37] and applied to the shape design of the $N + 2$ Supersonic Passenger Jet. Applications to enhance optimization methods have been developed in [68, 23, 20, 18]. AS has also been successfully used to reduce the memory consumption of highly parametrized systems such as artificial neural networks [17, 39].

Possible extensions and variants of the active subspaces property are the Local Active Subspace method [50], the Active Manifold method [10] which reduces the problem to the analysis of a 1D manifold by traversing the level sets of the model function at the expense of high online costs, the shared Active Subspace method [31], the active subspaces property for multivariate functions [75], and more recently an extension of AS to dynamical systems [5]. Another method is Nonlinear Level set Learning (NLL) [77] which exploits RevNets to reduce the input parameter space with a nonlinear transformation.

The search for low dimensional structures is also investigated in machine learning with manifold learning algorithms. In this context the Active Subspaces methodology can be seen as a supervised dimension reduction technique along with Kernel Principal Component Analysis (KPCA) [59] and Supervised Kernel Principal Component Analysis (SKPCA) [7]. Other methods in the context of kernel-based ROMs are [26, 32, 40]. In [43] a non-linear extension of the active subspaces property based on Random Fourier Features [47, 36] is introduced and compared with machine learning manifold learning algorithms for the construction of Gaussian process regressions (GPR) [73].

From the preliminary work [43] in the context of supervised dimension reduction algorithms in machine learning, we develop the kernel-based active subspaces (KAS) method. The novelties of our contribution are the following:

- regarding the AS theoretical background, we provide an upper bound of the ridge approximation error (2) for vector-valued objective functions and for a wide collection of probability distributions (see Assumption 3).
- we extend kernel-based AS to vector-valued model functions and develop a detailed algorithmic procedure for the optimization of the feature map. We also test different spectral measures (see Equation (3.2) for the definition), differently from [43] where only the Gaussian measure is employed.

¹Some authors refer to the active subspaces method, we prefer to employ the term active subspaces property, as suggested by Constantine.

- the application to several test problems of increasing complexity. In particular, we mainly test KAS on problems where the active subspace is not present or the behaviour is not linear, differently from [43], where the comparison is made with KPCA and its variants on datasets with linear trends in the reduced parameter space, apart from the hyperparaboloid test case that we have also included among our toy problems.
- the KAS method is finally applied to a computational fluid dynamics problem and compared with the standard AS technique. We study the evolution of fluid flow past a NACA 0012 airfoil in a duct composed by an initialization channel and a chamber. The motion is modelled with the unsteady incompressible Navier-Stokes equations, and discretized with the Discontinuous Galerkin method (DG) [28]. Physical and geometrical parameters are introduced and sensitivity analysis of the lift and drag coefficients with respect to these parameters is provided.

The work is divided as follows: in section 2 we briefly present the active subspaces property of a model function with a focus on the construction of Gaussian process response surfaces. Then, section 3 illustrates the novel method called kernel-based active subspaces for both scalar and vector-valued model functions. Several tests to compare AS and KAS are provided in section 4 where we start from scalar functions with radial symmetry, we analyze an epidemiology model and a vector-valued output generated from a stochastic elliptic PDE. A parametric CFD test case for the study of the flow past a NACA airfoil using the Discontinuous Galerkin method is presented in section 5. Finally, we outline some perspectives and future studies in section 6.

2 Active Subspaces for parameter space reduction

Active Subspaces (AS) approach proposed in [53] and developed in [12] is a technique for dimension reduction in parameter space. In brief AS are defined as the leading eigenspaces of the second moment matrix of the model function's gradient (for scalar model functions) and constitutes a global sensitivity index [75]. In the context of ridge approximation, the choice of the active subspace corresponds to the minimizer of an upper bound of the mean square error obtained through Poincaré-type inequalities [75]. After performing dimension reduction in the parameter space through AS, the method can be applied to reduce the computational costs of different parameter studies such as inverse problems, optimization tasks and numerical integration. In this work we are going to focus on the construction of response surfaces with Gaussian process regression.

Definition 1 (Hypothesis on input and output spaces). *The quantities related to the input space are:*

- $m \in \mathbb{N}$ the dimension of the input space,
- (Ω, \mathcal{F}, P) the probability space,
- $\mathbf{X} : (\Omega, \mathcal{F}, P) \rightarrow \mathbb{R}^m$, the absolutely continuous random vector representing the parameters,
- $\rho : \mathbb{R}^m \rightarrow \mathbb{R}$, the probability density of \mathbf{X} with support $\mathcal{X} \subset \mathbb{R}^m$.

The quantities related to the output are:

- $d \in \mathbb{N}$ the dimension of the output space,
- $V = (\mathbb{R}^d, R_V)$ the Euclidean space with metric $R_V \in \mathcal{M}(d \times d)^2$ and norm

$$\|\mathbf{x}\|_{R_V}^2 = \mathbf{x}^T R_V \mathbf{x},$$

- $f : \mathcal{X} \subset \mathbb{R}^m \rightarrow V$, the quantity/function of interest, also called objective function in optimization tasks.

Let $\mathcal{B}(\mathbb{R}^m)$ be the Borel σ -algebra of \mathbb{R}^m . We will consider the Hilbert space $L^2(\mathbb{R}^m, \mathcal{B}(\mathbb{R}^m), \rho; V)$, of the measurable functions $f : (\mathbb{R}^m, \mathcal{B}(\mathbb{R}^m), \rho) \rightarrow (\mathbb{R}^d, R_V)$ such that

$$\|f\|_{L^2}^2 := \int_{\mathcal{X}} \|f(\mathbf{x})\|_{R_V}^2 d\rho(\mathbf{x}) \leq \infty;$$

²In this work with $\mathcal{M}(m \times n)$ we denote the set of real matrices with m rows and n columns.

and the Sobolev space $H^1(\mathbb{R}^m, \mathcal{B}(\mathbb{R}^m), \rho; V)$ of measurable functions $f : (\mathbb{R}^m, \mathcal{B}(\mathbb{R}^m), \rho) \rightarrow (\mathbb{R}^d, R_V)$ such that

$$\|f\|_{H^1}^2 := \|f\|_{L^2}^2 + \|\nabla f\|_{L^2}^2 = \|f\|_{L^2}^2 + |f|_{H^1}^2 \leq \infty \quad (1)$$

where ∇f is the weak derivative of f , and $\|\nabla f\|_{L^2} =: |f|_{H^1}$.

We briefly recall how dimension reduction in parameter space is achieved in the construction of response surfaces. The first step involves the approximation of the model function with ridge approximation. We will follow [75, 44] for a review of the method.

The ridge approximation problem can be stated in the following way:

Definition 2 (Ridge approximation). *Let $\mathcal{B}(\mathbb{R}^m)$ be the Borel σ -algebra of \mathbb{R}^m . Given $r \in \mathbb{N}$, $r \ll d$ and a tolerance $\epsilon \geq 0$, find the profile $h : (\mathbb{R}^m, \mathcal{B}(\mathbb{R}^m), \rho) \rightarrow V$ and the r -rank projection $P_r : \mathbb{R}^m \rightarrow \mathbb{R}^m$ such that*

$$\mathbb{E}_P[\|f(\mathbf{X}) - h(P_r \mathbf{X})\|_{R_V}^2] \leq \epsilon^2. \quad (2)$$

In particular we are interested in the minimization problem

$$\operatorname{argmin}_{P_r \in \mathcal{M}(m \times m)} \mathbb{E}_P \left[\|f(\mathbf{X}) - \tilde{h}(P_r \mathbf{X})\|_{R_V}^2 \right], \quad (3)$$

where $\tilde{h} = \mathbb{E}_\rho[f|\sigma(P_r)]$ is the conditional expectation of f under the distribution ρ given the σ -algebra $\sigma(P_r)$. The range of the projector P_r , $\mathbb{R}^r \sim \operatorname{Im}(P_r) \subset \mathbb{R}^m$, is the reduced parameter space. The kernel of the projector P_r , $\mathbb{R}^{m-r} \sim \operatorname{Im}(P_r) \subset \mathbb{R}^m$, is the inactive subspace. The existence of \tilde{h} is guaranteed by the Doob-Dynkin lemma [9]. The function \tilde{h} is proven to be the optimal profile for each fixed P_r , as a consequence of the definition of the conditional expectation of a random variable with respect to a σ -algebra.

Dimension reduction is effective if the inequality (2) is satisfied for a specific tolerance. The choice of r is certainly of central importance. The dimension of the reduced parameter space can be chosen a priori for a specific parameter study (for example r -dimensional regression), it can be chosen in order to satisfy the inequality (2) or it is determined to guarantee a good accuracy of the numerical method used to evaluate it [Corollary 3.10, [14]].

Dividing the left term of the inequality (2) with $\mathbb{E}_P[\|f(\mathbf{X}) - \mathbb{E}_\rho[f(\mathbf{X})]\|_{R_V}^2]$ we obtain the Relative Root Mean Square Error (RRMSE) and since it is a normalized quantity, we will use it to make comparisons between different models

$$\operatorname{RRMSE} = \sqrt{\frac{\mathbb{E}_P[\|f(\mathbf{X}) - h(P_r \mathbf{X})\|_{R_V}^2]}{\mathbb{E}_P[\|f(\mathbf{X}) - \mathbb{E}_\rho[f(\mathbf{X})]\|_{R_V}^2]}}. \quad (4)$$

We remark that P_r is not unique. It can be shown that if \tilde{h} is the optimal profile, then P_r is not uniquely defined and can be chosen arbitrarily from the set $\{Q_r : \mathbb{R}^m \rightarrow \mathbb{R}^m \mid \ker Q_r = \ker P_r\}$, see [Proposition 2.2, [75]].

The following lemma is the key ingredient in the proof of the existence of an active subspace. It is inherently linked to probability Poincaré inequalities of the kind

$$\int_{\mathcal{X}} \|h(\mathbf{x})\|_{L^2}^2 d\rho(\mathbf{x}) \leq C_P(\mathcal{X}, \rho) \int_{\mathcal{X}} \|\nabla h(\mathbf{x})\|_{L^2}^2 d\rho(\mathbf{x}), \quad (5)$$

for zero-mean functions in the Sobolev space $h \in H^1(\mathcal{X})$, where $C_P(\mathcal{X}, \rho)$ is the Poincaré constant dependent on the domain \mathcal{X} and on the probability density functions (p.d.f.), ρ . We need to make the following assumption to prove the next lemma and the next theorem.

Definition 3. *The probability density function $\rho : \mathcal{X} \rightarrow \mathbb{R}$ belongs to one of the following classes:*

1. \mathcal{X} is convex and bounded, $\exists \delta, D > 0 : 0 < \delta \leq \|\rho(\mathbf{x})\|_{L^\infty} \leq D < \infty \forall \mathbf{x} \in \mathcal{X}$,
2. $\rho(\mathbf{x}) \sim \exp(-V(\mathbf{x}))$ where $V : \mathbb{R}^m \rightarrow (-\infty, \infty]$, $V \in \mathcal{C}^2$ is α -uniformly convex,

$$\mathbf{u}^T \operatorname{Hess}(V(\mathbf{x})) \mathbf{u} \geq \alpha \|\mathbf{u}\|_2^2, \quad \forall \mathbf{x}, \mathbf{u} \in \mathbb{R}^m \quad (6)$$

where $\operatorname{Hess}(V(\mathbf{x}))$ is the Hessian of $V(\mathbf{x})$.

3. $\rho(\mathbf{x}) \sim \exp(-V(\mathbf{x}))$ where V is a convex function. In this case we require also f Lipschitz continuous.

In particular the uniform distribution belongs to the first class, the multivariate Gaussian distribution $\mathcal{N}(m, \Sigma)$ to the second with $\alpha = 1/(\sigma_{max}(\Sigma))$ and the exponential and Laplace distributions to the third. A complete analysis of the various cases is done in [44].

Proposition 1. *Let (Ω, \mathcal{F}, P) be a probability space, $\mathbf{X} : (\Omega, \mathcal{F}, P) \rightarrow \mathbb{R}^m$ an absolutely continuous random vector with probability density function ρ belonging to one of the classes from the Assumption 3. Then the following inequality is satisfied*

$$\mathbb{E}_\rho \left[(h - \mathbb{E}_\rho[h|\sigma(P_r)])^2 | \sigma(P_r) \right] \leq C_P(P_r, \rho) \mathbb{E}_\rho \left[\|(I - P_r^T) \nabla h\|_2^2 | \sigma(P_r) \right] \quad (7)$$

for all scalar functions $h \in H^1(\mathcal{X})$ and for all r -rank orthogonal projectors, P_r , where $C_P(P_r, \rho)$ is the Poincaré constant depending on P_r and on the p.d.f. ρ .

A summary of the values of the Poincaré constant in relationship with the choice of the probability density function ρ is reported in [44].

In the next theorem the projection P_r will depend on the output function f , so also the Poincaré constant $C_P(P_r, \rho)$ will depend in fact on f .

We introduce the following notation for the matrix that substitutes the uncentered covariance matrix of the gradient ∇f in the case of the application of AS to scalar model functions [14]

$$H = \int_{\mathcal{X}} (D_{\mathbf{x}}f(\mathbf{x}))^T R_V(\rho) (D_{\mathbf{x}}f(\mathbf{x})) d\rho(\mathbf{x}).$$

where $D_{\mathbf{x}}f(\mathbf{x}) \in \mathcal{M}(d \times m)$ is the Jacobian matrix of f . The matrix $R_V(\rho)$ depends on the class which ρ belongs to, see Appendix A.

Theorem 1 (Existence of an active subspace). *Under the hypothesis 1, let $f \in H^1(\mathbb{R}^m, \mathcal{B}(\mathbb{R}^m), \rho; V)$ and let the p.d.f. ρ satisfy Lemma 1 and Assumption 3. Then the solution \tilde{P}_r of the ridge approximation problem 2 is the orthogonal projector to the eigenspace of the first r -eigenvalues of H ordered by magnitude*

$$Hv_i = \lambda_i v_i \quad \forall i \in \{1, \dots, m\}, \quad \tilde{P}_r = \sum_{j=1}^r v_j \otimes v_j,$$

with $r \in \mathbb{N}$ chosen such that

$$\mathbb{E}_\rho \left[\|f - \tilde{h}\|_{R_V}^2 \right] \leq C(C_P, \tau) \left(\sum_{i=r+1}^m \lambda_i \right)^{\frac{1}{1+\tau}} \leq \epsilon^2. \quad (8)$$

with $C(C_P, \tau)$ a constant depending on $\tau > 0$ related to the choice of ρ and on the Poincaré constant from lemma 1, and $\tilde{h} = \mathbb{E}_\rho[f|\sigma(P_r)]$ is the conditional expectation of f given the σ -algebra generated by the random variable $P_r \circ \mathbf{X}$.

Proof. This theorem summarizes the results from Proposition 2.5 and Proposition 2.6 of [75], and from Lemma 3.1, Lemma 4.2, Lemma 4.3, Lemma 4.4 and Theorem 4.5 of [44]. The proof is expanded in Appendix A. \square

The eigenspace $\text{span}\{v_1, \dots, v_r\} \subset \mathbb{R}^m$ is the active subspace and the remaining eigenvectors generate the inactive subspace $\text{span}\{v_{r+1}, \dots, v_m\} \subset \mathbb{R}^m$. The condition $f \in L^2(\mathbb{R}^m, \mathcal{B}(\mathbb{R}^m), \rho; V)$ is necessary for f to satisfy the error bound (2).

For the explicit procedure to compute the active subspace given its dimension r see Algorithm 1: from W_1 and W_2 we define the approximations of the projector P_r with $\hat{P}_r = W_1 W_1^T$.

Algorithm 1 Active subspace computation.

Require: gradients dataset $dY = (dy_1, \dots, dy_M)^T$, $dy_i \in \mathcal{M}(d \times m)$, that is dY is a 3-rank tensor

Require: symmetric positive definite metric matrix $R_V \in \mathcal{M}(d \times d)$

Require: active subspace dimension r

- 1: Compute the uncentered covariance matrix with Monte Carlo:

$$\tilde{H} = \frac{1}{M} \sum_{j=1}^M dY[j, :, :]^T R_V dY[j, :, :].$$

- 2: Solve the eigenvalue problem:

$$\begin{aligned} \tilde{H} \mathbf{v}_i &= \lambda_i \mathbf{v}_i \quad \forall i \in \{1, \dots, m\}, \\ W_1 &= (\mathbf{v}_1, \dots, \mathbf{v}_r), \quad W_2 = (\mathbf{v}_{r+1}, \dots, \mathbf{v}_m) \Rightarrow \hat{P}_r := W_1 W_1^T. \end{aligned}$$

- 3: **return** active eigenvectors $W_1 = (\mathbf{v}_1, \dots, \mathbf{v}_r)$ and inactive eigenvectors $W_2 = (\mathbf{v}_{r+1}, \dots, \mathbf{v}_m)$ with $\mathbf{v}_i \in \mathbb{R}^m$, and ordered eigenvalues $(\lambda_1, \dots, \lambda_m)$
-

2.1 Response surfaces

The term response surface refers to the general procedure of finding the values of a model function f for new inputs without directly computing it but exploiting regression or interpolation from a training set $\{\mathbf{x}_i, f(\mathbf{x}_i)\}$. The procedure for constructing a Gaussian process response is reported in Algorithm 2, while in Algorithm 3 we show how to exploit it to predict the model function at new input parameters.

Directly applying the simple Monte Carlo method with N samples we get a reduced approximation of f as

$$(\tilde{h}_\epsilon \circ P_r)(\mathbf{X}) = \mathbb{E}_\rho [f | \sigma(P_r)] \approx \frac{1}{N} \sum_{i=1}^N f(\hat{P}_r \mathbf{X} + (I_d - \hat{P}_r) \mathbf{Y}_i) =: \hat{h}_{\epsilon, N}(\hat{P}_r \mathbf{X}), \quad (9)$$

where we have made explicit the dependence of the optimal profile \tilde{h}_ϵ on ϵ , $\mathbf{Y}_1, \dots, \mathbf{Y}_N$ are independent and identically distributed samples of $\mathbf{Y} \sim \rho$, and \hat{P}_r is an approximation of P_r obtained with the simple Monte Carlo method from H , see Algorithm 1. An intermediate approximation error is obtained employing the Poincaré inequality and the central limit theorem for the Monte Carlo approximation

$$\mathbb{E}_P \left[(f(\mathbf{X}) - \hat{h}_{\epsilon, N}(\hat{P}_r \mathbf{X}))^2 \right] \leq C_1 \left(1 + N^{-1/2} \right)^2 (\lambda_{n+1} + \dots + \lambda_m), \quad (10)$$

where C_1 is a constant, and $\lambda_{n+1}, \dots, \lambda_m$ are the eigenvalues of the inactive subspace of H [Theorem 4.4, [14]].

In practice $\hat{h}_{\epsilon, N}(\hat{P}_r \mathbf{X})$ is approximated with a regression or an interpolation such that a response surface \mathcal{R} satisfying $\mathbb{E}_\rho \left[(\hat{h}_{\epsilon, N}(\hat{P}_r \mathbf{x}) - \mathcal{R}(\hat{P}_r \mathbf{x}))^2 \right] \leq C_2 \delta$ is built, where C_2 is a constant, and δ depends on the chosen method. An estimate for the successive approximations

$$f(\mathbf{X}) \approx \tilde{h}_\epsilon(P_r \mathbf{X}) \approx \hat{h}_{\epsilon, N}(\hat{P}_r \mathbf{X}) \approx \mathcal{R}_{\epsilon, N, \delta}(\hat{P}_r \mathbf{X}), \quad (11)$$

is given by

$$\begin{aligned} & \mathbb{E}_P \left[(f(\mathbf{X}) - \mathcal{R}(\hat{P}_r \mathbf{X}))^2 \right] \\ & \leq C_1 (1 + N^{-1/2})^2 \left(\tau (\lambda_1 + \dots + \lambda_n)^{1/2} + (\lambda_{n+1} + \dots + \lambda_m)^{1/2} \right)^2 + C_2 \lambda \end{aligned}$$

where $\text{dist}(\text{Im}(P_r), \text{Im}(\hat{P}_r)) \leq \tau$, and λ_i are the eigenvalues of H [Theorem 4.8, [14]].

In our numerical simulations we will build the response surface \mathcal{R} with Gaussian process regression (GPR) [73].

Algorithm 2 Response surface construction with Gaussian process regression over the active subspace.

Require: normalized input dataset $X = (\mathbf{x}_1, \dots, \mathbf{x}_M)^T$, $\mathbf{x}_i \in \mathbb{R}^m$

Require: output dataset $Y = (y_1, \dots, y_M)^T$, $y_i \in \mathbb{R}$

Require: active eigenvectors $W_1 = (\mathbf{v}_1, \dots, \mathbf{v}_r)$, $\mathbf{v}_i \in \mathbb{R}^m$

Require: kernel $k : \mathbb{R}^m \times \mathbb{R}^m \rightarrow \mathbb{R}$, with hyper-parameters θ and the variance ϵ of the Gaussian noise

- 1: Project the inputs in the active subspace: $XW_1 = \tilde{X} \in \mathcal{M}(M \times r)$.
- 2: Evaluate the Gram matrix: $K_{ij}(\theta) = k(\tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}_j; \theta)$, $1 \leq i, j \leq r$.
- 3: Tune the hyperparameters minimizing the negative log-likelihood:

$$\bar{\theta} = \arg \min_{\theta} -\log p(y|x, \theta) \propto \frac{1}{2} \log |K(\theta) + \sigma I_M| + \frac{1}{2} Y^T (K(\theta) + \sigma I_M)^{-1} Y,$$

- 4: **return** trained Gaussian process
-

Algorithm 3 Prediction phase using the Gaussian process response surface over the active subspace.

Require: trained response surface $\mathbf{y}(\mathbf{x})$

Require: active eigenvectors $W_1 = (\mathbf{v}_1, \dots, \mathbf{v}_r)$, $\mathbf{v}_i \in \mathbb{R}^m$

Require: test samples $\bar{\mathbf{x}} \in \mathbb{R}^m$

- 1: Map the test samples $\bar{\mathbf{x}}$ onto the active subspace: $\tilde{\mathbf{x}} = W_1 \bar{\mathbf{x}}$.
- 2: Evaluate the Gaussian process on $\tilde{\mathbf{x}}$ and return the prediction $t \sim \mathcal{N}(\mathbb{E}[t], \sigma^2(t))$:

$$\mathbb{E}[t] = k(\tilde{\mathbf{x}}, X)K^{-1}Y, \quad \sigma^2(t) = k(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}) - k(\tilde{\mathbf{x}}, X)K^{-1}k(X, \tilde{\mathbf{x}}).$$

3 Kernel-based Active Subspaces extension

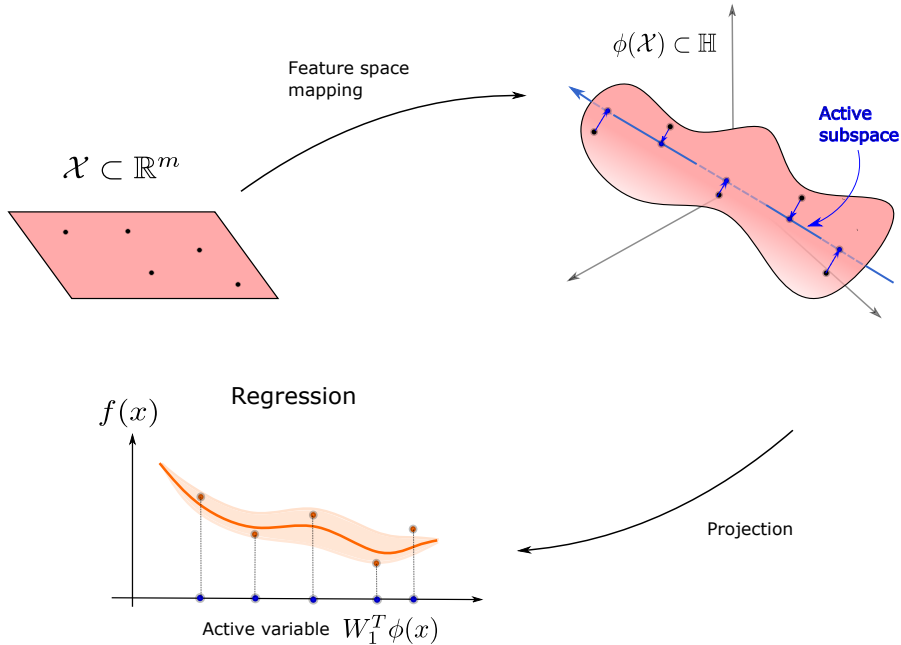


Figure 1: Illustration of the construction of a one-dimensional response surface with kernel-based active subspaces and Gaussian process regression.

Keeping the notations of section 1, $\mathbf{X} : (\Omega, \mathcal{F}, P) \rightarrow \mathbb{R}^m$ is the absolutely continuous random vector representing the m -dimensional inputs with density $\rho : \mathcal{X} \subset \mathbb{R}^m \rightarrow \mathbb{R}$, and $f : \mathcal{X} \subset \mathbb{R}^m \rightarrow (V, R_V)$ is the model function that we assume to be continuously differentiable and Lipschitz

continuous.

One drawback of sufficient dimension reduction with AS applied to ridge approximation is that if a clear linear trend is missing, projecting the inputs as $P_r \mathbf{X}$ represents a loss of accuracy on the approximation of the model f that may not be compensated even by the choice of the optimal profile $\hat{h} \circ P_r = \mathbb{E}_\rho[f|\sigma(P_r)]$. In order to overcome this, non-linear dimension reduction to one-dimensional parameter space could be achieved discovering a curve in the space of parameters that cuts transversely the level sets of f , this variation is presented in [10] as Active Manifold. Another approach could consist in finding a diffeomorphism ϕ that reshapes the level sets such that subsequently applying AS dimension reduction to the new model function $\tilde{f} \circ \phi = f$ could be more profitable:

$$\begin{array}{ccc} \mathcal{X} \subset \mathbb{R}^m & \xrightarrow{\phi} & \phi(\mathcal{X}) \subset \mathbb{R}^m \\ & \searrow f & \downarrow \tilde{f} \\ & & V \end{array}$$

Unfortunately constructing the Active Manifold or finding the right diffeomorphism ϕ could be a complicated matter. If we renounce to have a backward map and we weaken the bond of the method with the model, we can consider an immersion ϕ from the space of parameters \mathcal{X} to an infinite-dimensional Hilbert space \mathbb{H} obtaining

$$\begin{array}{ccc} \mathcal{X} \subset \mathbb{R}^m & \xrightarrow{\phi} & \phi(\mathcal{X}) \subset \mathbb{H} \\ & \searrow f & \downarrow \tilde{f} \\ & & V \end{array}$$

This is a common procedure in machine learning in order to increase the number of features [73]. Then AS is applied to the new model function $\tilde{f} : \phi(\mathcal{X}) \subset \mathbb{H} \rightarrow V$ with parameter space $\phi(\mathcal{X}) \subset \mathbb{H}$. A response surface can be built with 2 remembering to replace every occurrence of the inputs \mathbf{x} with their images $\phi(\mathbf{x})$. A synthetic scheme of the procedure is represented in Figure 1.

In practice we consider a discretization of the infinite-dimensional Hilbert space $\mathbb{R}^D \simeq \mathbb{H}$ with $D > m$. Dimension reduction with AS results in the choice of a r -rank projection in the much broader set of r -rank projections in \mathbb{H} .

Since for AS only the samples of the Jacobian matrix of the model function are employed, we can ignore the definition of the new map $\tilde{f} : \phi(\mathcal{X}) \subset \mathbb{H} \rightarrow (V, R_V)$ and focus only on the computation of the Jacobian matrix of \tilde{f} with respect to the new input variable $\mathbf{z} := \phi(\mathbf{x})$. The uncentered covariance matrix becomes

$$\begin{aligned} H &= \int_{\phi(\mathcal{X})} \left[(D_{\mathbf{z}} \tilde{f})^T(\mathbf{z}) \right] R_V \left[(D_{\mathbf{z}} \tilde{f})(\mathbf{z}) \right] d\mu(\mathbf{z}) \\ &= \int_{\mathcal{X}} \left[(D_{\mathbf{z}} \tilde{f})^T(\phi(\mathbf{x})) \right] R_V \left[(D_{\mathbf{z}} \tilde{f})(\phi(\mathbf{x})) \right] d\mathcal{L}_{\mathbf{X}}(\mathbf{x}), \end{aligned}$$

where $\mu := \phi_{\#}(\mathcal{L}_{\mathbf{X}})$ is the pushforward probability measure of $\mathcal{L}_{\mathbf{X}}$ (the law of probability of \mathbf{X}) with respect to the map ϕ . Simple Monte Carlo can be applied sampling from the distribution ρ in the input space \mathcal{X}

$$\begin{aligned} H &= \int_{\mathcal{X}} \left[(D_{\mathbf{z}} \tilde{f})^T(\phi(\mathbf{x})) \right] R_V \left[(D_{\mathbf{z}} \tilde{f})(\phi(\mathbf{x})) \right] d\mathcal{L}_{\mathbf{X}}(\mathbf{x}) \\ &\approx \frac{1}{M} \sum_{i=1}^M \left[(D_{\mathbf{z}} \tilde{f})^T(\phi(\mathbf{x}_i)) \right] R_V \left[(D_{\mathbf{z}} \tilde{f})(\phi(\mathbf{x}_i)) \right]. \end{aligned}$$

The gradients of \tilde{f} with respect to the new input variable \mathbf{Z} are computed from the known values $D_{\mathbf{x}} f$ with the chain rule.

The application of the chain rule to the composition of functions $\tilde{f} \circ \phi : \mathbb{R}^m \rightarrow \mathbb{H} \rightarrow V$ is applicable if \tilde{f} is defined in an open set $U \supset \phi(\mathcal{X})$. If ϕ is non singular and also injective the new input space is a m -dimensional submanifold of \mathbb{H} . If ϕ is also smooth there exists a smooth extension of $\tilde{f} : \phi(\mathcal{X}) \subset \mathbb{H} \rightarrow V$ onto the whole domain \mathbb{H} , see Proposition 1.36 from [71].

If the Hilbert space \mathbb{H} has finite dimension $\mathbb{H} \sim \mathbb{R}^D$ this procedure leaves us with an underdetermined linear system to solve for $D_{\mathbf{z}}\tilde{f}$

$$\begin{aligned} D_{\mathbf{z}}\tilde{f}(\phi(\mathbf{x}))D\phi(\mathbf{x}) &= D_{\mathbf{x}}f(\mathbf{x}), \\ D_{\mathbf{z}}\tilde{f}(\phi(\mathbf{x})) &= D_{\mathbf{x}}f(\mathbf{x})(D\phi(\mathbf{x}))^\dagger, \end{aligned} \quad (12)$$

where † stands for the right Moore-Penrose inverse of the matrix $D\phi(\mathbf{x})$ with rank r , that is

$$(D\phi(\mathbf{x}))^\dagger = V\Sigma^\dagger U^T,$$

with the usual notation for the singular value decomposition (SVD) of $D\phi(\mathbf{x})$

$$D\phi(\mathbf{x}) = U\Sigma V^T, \quad (13)$$

and $\Sigma^\dagger \in \mathcal{M}(r \times r)$ equal to the diagonal matrix with the inverse of the singular values as diagonal elements. As anticipated if f is smooth enough and ϕ is an embedding, so that $D\phi$ has full rank, the previous system has a unique solution. The most crucial part is the evaluation of the gradients $D_{\mathbf{x}}f(\mathbf{x})$ from the input output couples, when they are not available analytically or from the adjoint method applied to PDEs models: different approaches are present in the literature, like local polynomial regressions and Gaussian process regression on the whole domain to approximate the gradients; both are available in the ATHENA package [49]. For an estimate of the ridge approximation error due to inaccurate gradients see [12].

Finally, we remark that in the AS method we approximate the random variable \mathbf{X} as

$$P_r\mathbf{X} = \mathbf{v}_1(\mathbf{v}_1 \cdot \mathbf{X}) + \dots + \mathbf{v}_r(\mathbf{v}_r \cdot \mathbf{X}), \quad (14)$$

with $\{\mathbf{v}_i\} \subset \mathbb{R}^m$ the active eigenvectors, whereas with KAS the reduced input space is contained in \mathcal{H}

$$P_r\mathbf{X} = \mathbf{v}_1(\mathbf{v}_1 \cdot \phi(\mathbf{X})) + \dots + \mathbf{v}_r(\mathbf{v}_r \cdot \phi(\mathbf{X})), \quad (15)$$

with $\{\mathbf{v}_i\} \subset \mathcal{H}$ the active eigenvectors of KAS. In this case the model is enriched by the non-linear feature map ϕ .

Algorithm 4 Kernel-based active subspace computation.

Require: gradients dataset $dY = (dy_1, \dots, dy_M)^T$, $dy_i \in \mathcal{M}(d \times m)$

Require: symmetric positive definite metric matrix $R_V \in \mathcal{M}(d \times d)$

Require: feature subspace dimension D

Require: feature map $\phi : \mathbb{R}^m \rightarrow \mathbb{R}^D$

Require: active subspace dimension r

- 1: Evaluate gradients solving an overdetermined linear system:

$$\forall j \in \{1, \dots, M\}, \quad dY[j, :, :](D\phi)^\dagger = d\tilde{Y}[j, :, :] \in \mathcal{M}(d, D).$$

- 2: Compute the uncentered covariance matrix with Monte Carlo:

$$\tilde{H} = \frac{1}{M} \sum_{k=1}^M d\tilde{Y}[k, :, :]^T R_V d\tilde{Y}[k, :, :].$$

- 3: Solve the eigenvalue problem: $\tilde{H}\mathbf{v}_i = \lambda_i\mathbf{v}_i$, $\forall i \in \{1, \dots, D\}$.

- 4: **return** active eigenvectors $W_1 = (\mathbf{v}_1, \dots, \mathbf{v}_r)$ and inactive eigenvectors $W_2 = (\mathbf{v}_{r+1}, \dots, \mathbf{v}_D)$ with $\mathbf{v}_i \in \mathbb{R}^D$, and ordered eigenvalues $(\lambda_1, \dots, \lambda_D)$
-

3.1 Choice of the Feature Map

The choice for the map ϕ is linked to the theory of Reproducing Kernel Hilbert Spaces (RKHS) [8], and it is defined as

$$\mathbf{z} = \phi(\mathbf{x}) = \sqrt{\frac{2}{D}} \sigma_f \cos(W\mathbf{x} + \mathbf{b}), \quad (16)$$

$$\cos(W\mathbf{x} + \mathbf{b}) := \frac{1}{\sqrt{D}} (\cos(W[1, :] \cdot \mathbf{x} + b_1), \dots, \cos(W[D, :] \cdot \mathbf{x} + b_D))^T \quad (17)$$

where σ_f is an hyperparameter corresponding to the empirical variance of the model, $W \in \mathcal{M}(D \times m)$ is the projection matrix whose rows are sampled from a probability distribution μ on \mathbb{R}^m and $\mathbf{b} \in \mathbb{R}^D$ is a bias term whose components are sampled independently and uniformly in the interval $[0, 2\pi]$. We remark that its Jacobian can be computed analytically as

$$\frac{\partial z^j}{\partial x^i} = -\sqrt{\frac{2}{D}} \sigma_f \sin\left(\sum_{k=1}^D W_{ik} \mathbf{x}_k + \mathbf{b}_k\right) W_{ij}, \quad (18)$$

for all $i \in \{1, \dots, m\}$, and for all $j \in \{1, \dots, D\}$.

We remark that in order to guarantee the correctness of the procedure for evaluating the gradients we have to prove that the feature map is injective and non singular. In general however the feature map (16) cannot not be injective due to the periodicity of the cosine but at least it is almost surely non singular if the dimension of the feature space is high enough.

The feature map (16) is not the only effective immersion that provides a kernel-based extension of the active subspaces. For example an alternative is the following composition of a linear map with a sigmoid

$$\phi(\mathbf{z}) = \frac{C}{1 + \alpha e^{-W\mathbf{z}}},$$

where C is a constant, α is an hyperparameter to be tuned, and $W \in \mathcal{M}(D, m)$ is, as before, a matrix whose rows are sampled from a probability distribution on \mathbb{R}^m .

Other choices involve the use of Deep Neural Networks to learn the profile h and the projection function P_r of the ridge approximation problem [67].

The tuning of the hyperparameters of the spectral measure consists in a global optimization problem where the dimension of the domain can vary between 1 and the dimension of the input space m . The object function to optimize is the relative root mean square error (RRMSE)

$$\text{RRMSE}(Y_{\text{test}}, T_{\text{test}}) = \sqrt{\frac{\sum_{i=1}^N (t_i - y_i)^2}{\sum_{i=1}^N (t_i - \bar{y})^2}}, \quad (19)$$

where $T_{\text{test}} = (t_i)_{i \in \{1, \dots, N\}}$ are the predictions obtained from the response surface built with KAS and associated to the test set, $Y_{\text{test}} = (y_i)_{i \in \{1, \dots, N\}}$ are the targets associated to the test set, and \bar{y} is the mean value of the targets. We implemented a logarithmic grid-search, see Algorithm 5, making use of the SciPy library [69]. Another choice could be Bayesian stochastic optimization implemented in the open-source library GPyOpt [3].

The tuning of the hyperparameters of the spectral measure chosen is the most computationally expensive part of the procedure. We report the computational complexity of the algorithms introduced to have a better understanding of the additional cost implied by the implementation of response surface design with KAS. Let us assume that the number of random Fourier features D , the number of input, output, and gradient samples M , and the dimension of the parameter space m , are ordered in this manner $D > M > m$, as is usually the case, and that the quantity of interest f is a scalar function. The cost of computing an active subspace is $O(Mm^2)$, that is the cost of the SVD of the gradients matrix dY used to get the active and inactive eigenvectors in Algorithm 1. The cost of the training of a response surface with Gaussian process regression in Algorithm 2 depends on the cost of minimization of the log-likelihood: each evaluation of the log-likelihood involves the computation of the determinant and the inverse of the regularized Gram matrix $K(\theta) + \sigma I_M$, that is $O(M^3)$. Finally, the cost for the evaluation of the kernel-based active subspace is associated to the SVD of $d\tilde{Y}$ that is $O(DM^2)$ in Algorithm 4, and to the resolution of the overdetermined linear system to obtain the gradients $d\tilde{Y}$, that is M times $O(Dm^2)$ since it is related to the evaluation of the pseudo-inverse of $D\phi$. So, the computational complexity for the response surface design with AS and GPR is $O(n_{\text{GPR}}M^3)$, while for the response surface design with KAS and GPR is $O\left(n_{\text{grid-search}}n\left(D\frac{M^2}{n^2} + \frac{M}{n}Dm^2 + n_{\text{GPR}}\frac{M^3}{n^3}\right)\right)$, where n_{GPR} is the maximum number of steps of the optimization algorithm used to minimize the log-likelihood, $n_{\text{grid-search}}$ is the number of hyperparameter instances $\gamma \in G$ to try in Algorithm 5, and n is the number of batches in the n -fold cross validation procedure. In particular, for each grid search hyperparameter the main cost is associated to the GPR training since n_{GPR} usually satisfy $Dn < n_{\text{GPR}}M$, when the optimizer chosen is L-BFGS-B from SciPy [69], accounting also for the number of restarts of the optimizer: in the numerical tests we performed the number of restarts of the training of the

GPR is problem-dependent but always less than 10. In general, the number $n_{\text{grid-search}}$ depends on the chosen application, and the multiplicative factor between the computational complexity of the response surface design procedure with KAS or AS is lower than $3n_{\text{grid-search}}n$.

Algorithm 5 Tuning the feature map with logarithmic grid-search.

Require: normalized input dataset $X = (\mathbf{x}_1, \dots, \mathbf{x}_M)^T$, $\mathbf{x}_i \in \mathbb{R}^m$

Require: output dataset $Y = (y_1, \dots, y_M)^T$, $y_i \in \mathbb{R}$

Require: gradients dataset $dY = (dy_1, \dots, dy_M)^T$, $dy_i \in \mathcal{M}(d \times D)$

Require: spd metric matrix $R_V \in \mathcal{M}(d \times d)$

Require: feature subspace dimension D

Require: feature map $\phi : \mathbb{R}^m \rightarrow \mathbb{R}^D$

Require: spectral density with hyperparameter α , $\mu = \mu(\alpha)$

Require: active subspace dimension r

Require: tolerance for the tuning procedure $\text{tol} \approx 0.8$.

- 1: Create the grid G and set the variable BEST to 1.
 - 2: **for** $\gamma \in G$ **do**
 - 3: Compute the feature map projection matrix W associated to γ : $W[i, :]$ sampled from $\mu(\gamma)$, $\forall i \in \{1, \dots, D\}$.
 - 4: Compute the uniformly sampled bias b : $b[i] \sim \mathcal{U}(0, 2\pi)$.
 - 5: Compute score with n -fold cross validation:
 - 6: **for** $i = 1$ to n **do**
 - 7: Divide input, output, and gradients in train and test datasets.
 - 8: Compute $(W_1, W_2, (\lambda_1, \dots, \lambda_D))$ with KAS method in Algorithm 4 with inputs $(dY_{\text{train}}, R_V, D, \phi, r)$.
 - 9: Build GPR response surface with inputs $(X_{\text{train}}, Y_{\text{train}}, W_1, k)$ using Algorithm 2.
 - 10: Predict the values T_{test} using Algorithm 3 with input X_{test} .
 - 11: Evaluate the score as $\text{score}[n] = \text{RRMSE}(Y_{\text{test}}, T_{\text{test}})$.
 - 12: **if** $\text{score}[n] > \text{tol}$ **then**
 - 13: Stop cross validation and pass to the next value of α .
 - 14: **end if**
 - 15: **end for**
 - 16: **if** $\text{mean}(\text{score}) < \text{BEST}$ **then**
 - 17: Save W and b , and set BEST to $\text{mean}(\text{score})$.
 - 18: **end if**
 - 19: **end for**
 - 20: **return** projection matrix W , and bias b
-

3.2 Random Fourier Features

The motivation behind the choice for this map from Equation (16) comes from the theory on Reproducing Kernel Hilbert Spaces. The infinite-dimensional Hilbert space $(\mathbb{H}, \langle \cdot, \cdot \rangle)$ is assumed to be a RKHS with real shift-invariant kernel $k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ with $k(0) = 1$ and feature map ϕ .

In order to get a discrete approximation of $\phi : \mathcal{X} \subset \mathbb{R}^m \rightarrow \mathbb{H}$, random Fourier features are employed [47, 36]. Bochner's theorem [41] guarantees the existence of a spectral probability measure μ such that

$$k(\mathbf{x}, \mathbf{y}) = \int_{\mathbb{R}^m} e^{i\boldsymbol{\omega} \cdot (\mathbf{x} - \mathbf{y})} d\mu(\boldsymbol{\omega}).$$

From this identity we can get a discrete approximation of the scalar product $\langle \cdot, \cdot \rangle$ with Monte Carlo method, exploiting the fact that the kernel is real

$$\langle \phi(\mathbf{x}), \phi(\mathbf{y}) \rangle = k(\mathbf{x}, \mathbf{y}) \approx \frac{1}{D} \sum_{i=1}^D \cos(\boldsymbol{\omega}_i \cdot \mathbf{x} + b_i) \cos(\boldsymbol{\omega}_i \cdot \mathbf{y} + b_i) = \mathbf{z}^T \mathbf{z}, \quad (20)$$

$$\mathbf{z} = \frac{1}{\sqrt{D}} (\cos(\boldsymbol{\omega}_1 \cdot \mathbf{x} + b_1), \dots, \cos(\boldsymbol{\omega}_D \cdot \mathbf{x} + b_D)), \quad (21)$$

and from this relation we obtain the approximation $\phi \approx \mathbf{z}$. The sampled vectors $\{\boldsymbol{\omega}_i\}_{i=1,\dots,D}$ are called random Fourier features. The scalars $\{b_i\}_{i=1,\dots,D}$ are bias terms introduced since in the approximation we have excluded some trigonometric terms from the following initial expression

$$\frac{1}{D} \sum_{i=1}^D (\cos(\boldsymbol{\omega}_i \cdot \mathbf{x}) \cos(\boldsymbol{\omega}_i \cdot \mathbf{y}) - \sin(\boldsymbol{\omega}_i \cdot \mathbf{x}) \sin(\boldsymbol{\omega}_i \cdot \mathbf{y})).$$

Random Fourier features are frequently used to approximate kernels. We consider only spectral probability measures which have a probability density, usually named spectral density. In the approximation of the kernel with random Fourier features, under some regularity conditions on the kernel, an explicit probabilistic bound depending on the dimension of the feature space D can be proved [41]. This technique is used to scale up Kernel Principal Component Analysis [56, 55] and Supervised Kernel Principal Component Analysis [7], but in the case of kernel-based AS the resulting overdetermined linear system employed to compute the Jacobian matrix of the new model function increases in dimension instead.

The most famous kernel is the squared exponential kernel also called Radial Basis Function kernel (RBF)

$$k_{\text{RBF}}(\mathbf{x}, \mathbf{y}) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{y}\|^2}{2l^2}\right), \quad (22)$$

where l is the characteristic length-scale. The spectral density is Gaussian $\mathcal{N}(0, 1/4\pi^2l^2)$:

$$S(\boldsymbol{\omega}) = (2\pi l^2)^{D/2} \exp(-2\pi^2 l^2 \boldsymbol{\omega}^2). \quad (23)$$

Thanks to Bochner’s theorem to every probability distribution that admits a probability density function corresponds a stationary positive definite kernel. So having in mind the definition of the feature map ϕ from Equation (16), we can choose any probability distribution for sampling the random projection matrix $W \in \mathcal{M}(D \times m)$ without focusing on the corresponding kernel since it is not needed by the numerical procedure.

After the choice of the spectral measure the corresponding hyperparameters have to be tuned. This is linked to the choice of the hypothesis models in machine learning and it is usually carried out for the hyperparameters of the employed kernel. From the choice of the kernel and the corresponding hyperparameters some regularity properties of the model are implicitly assumed [73].

4 Benchmark test problems

In this section we are going to present some benchmarks to prove the potential gain of KAS over standard linear AS, for both scalar and vectorial model functions. In particular we test KAS on radial symmetric functions, with 2-dimensional and 8-dimensional parameter spaces, on the approximation of the reproduction number R_0 of the SEIR model, and finally on a vectorial output function that is the solution of a Poisson problem.

One dimensional response surfaces are built following the algorithm described in subsection 2.1. The tuning of the hyperparameters of the feature map is carried out with a logarithmic grid-search and 5-fold cross validation for the Ebola test case, while for the other cases we employed Bayesian stochastic optimization implemented in [24] with 3-fold cross validation. The score function chosen is the Relative Root Mean Square Error (RRMSE). The spectral measure for each test case is chosen by brute force among the Laplace, Gaussian, Beta and multivariate Gaussian distributions. The number of Fourier features is not established based on a criterion but we have seen experimentally that above a certain threshold the number of features is high enough to at least reproduce the accuracy of the AS method. Since the most sensitive part to the final accuracy of the response surface is the tuning of the hyperparameters of the spectral measures, we suggest to choose an affordable number of features between 1000 and 2000, and focus on the tuning of said hyperparameters instead. We remark that the number of samples employed is problem dependent: some heuristics to determine it can be found in [12], but the crucial point is that additional training samples with respect to the ones used for the AS method are not needed for the novel KAS method. Moreover, the CPU time for the hyperparameters tuning procedure is usually negligible with respect to the time required to obtain input-output pairs from the numerical simulation of PDEs models: in our applications the tuning procedure’s computational time is in the order of minutes (usually around 10-15 minutes for most testcases), while for the CFD application of Section 5 it is

in the order of days and for the stochastic elliptic partial differential equation of Subsection 4.3 it is in the order of hours. We also remark that the tuning Algorithm 5, the GPR training restarts, and the choice of the spectral measure can be easily parallelized.

For the radial symmetric and Ebola test cases the inputs are sampled from a uniform distribution with problem dependent ranges. For the stochastic elliptic partial differential case the inputs are the coefficients of a Karhunen-Loève expansion and are sampled from a normal distribution. All the computations regarding AS and KAS are done using the open source Python package called ATHENA [49].

4.1 Radial symmetric functions

Radial symmetric functions represent a class of model functions for which AS is not able to unveil any low dimensional behaviour. In fact for these functions any rotation of the parameter space produce the same model representation. Instead Kernel-based AS is able to overcome this problem thanks to the mapping onto the feature space.

We present two benchmarks: an 8-dimensional hyperparaboloid defined as

$$f : [-1, 1]^8 \subset \mathbb{R}^8 \rightarrow \mathbb{R}, \quad f(\mathbf{x}) = \frac{1}{2} \|\mathbf{x}\|^2, \quad (24)$$

and the surface of revolution in \mathbb{R}^3 with generatrix $g(x) = \sin(x^2)$

$$f : [-3, 3]^2 \subset \mathbb{R}^2 \rightarrow \mathbb{R}, \quad f(\mathbf{x}) = g(\|\mathbf{x}\|) = \sin(\|\mathbf{x}\|^2). \quad (25)$$

The gradients are computed analytically.

For the hyperparaboloid we use $N_s = 500$ independent, uniformly distributed training samples in $[-1, 1]^8$, while for the sine case the training samples are $N_s = 800$ in $[-3, 3]^2$. In both cases the test samples are 500. The feature space has dimension 1000 for both the first and the second case. The spectral distribution chosen is the multivariate normal with hyperparameter a uniform variance λI_d , and a product of Laplace distributions with γ and b as hyperparameters, respectively. The tuning is carried out with 3-fold cross validation. The results are summarized in Table 1.

Table 1: Performance results for AS and KAS methods. For each case we report the parameter space dimension, the number of samples N_s used for the training, the chosen distribution, the dimension of the feature space, and the RRMSE mean and standard deviation for AS and KAS. In bold the best results.

Case	Dim	N_s	Spectral distribution	Feature space dim	RRMSE AS	RRMSE KAS
Hyperparaboloid	8	500	$\mathcal{N}(\mathbf{0}, \lambda I_d)$	1000	0.98 ± 0.03	0.23 ± 0.02
Sine	2	800	Laplace(γ, b)	1000	1.011 ± 0.01	0.31 ± 0.06
Ebola	8	800	Beta(α, β)	1000	0.46 ± 0.31	0.31 ± 0.03
SPDE (30)	10	1000	$\mathcal{N}(\mathbf{0}, \Sigma)$	1500	0.611 ± 0.001	0.515 ± 0.013

Looking at the eigenvalues of the uncentered covariance matrix of the gradients \tilde{H} for the hyperparaboloid case in Figure 2, we can clearly see how the decay for AS is almost absent, while using KAS the decay after the first eigenvalue is pronounced, suggesting the presence of a kernel-based active subspace of dimension 1.

The one-dimensional sufficient summary plots, which are $f(\mathbf{x})$ against $W_1^T \mathbf{x}$ — in the AS case — or against $W_1^T \phi(\mathbf{x})$ — in the KAS case —, are shown in Figure 3 and Figure 4, respectively. On the left panels we present the Gaussian process response surfaces obtained from the active subspaces reduction, while on the right panels the ones obtained with the kernel-based AS extension. As we can see AS fails to properly reduce the parameter spaces, since there are no preferred directions over which the model functions vary the most. The KAS approach, on the contrary, is able to unveil the corresponding generatrices. This results in a reduction of the RMS by a factor of at least 3 (see Table 1).

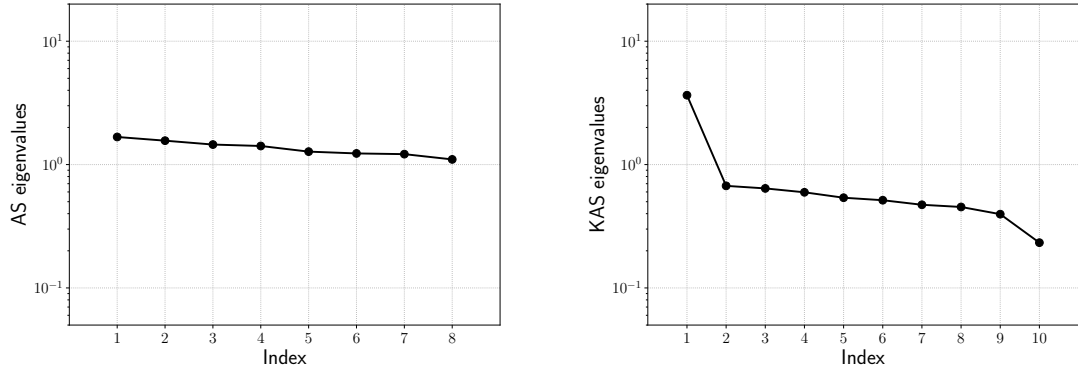


Figure 2: Eigenvalues of the covariance matrix $\tilde{H} \in \mathbb{R}^{8 \times 8}$ applied to the hyperparaboloid case for the AS procedure on the left, and the first 10 eigenvalues of the covariance matrix $\tilde{H} \in \mathbb{R}^{1000 \times 1000}$ for the KAS procedure applied to the same case on the right.

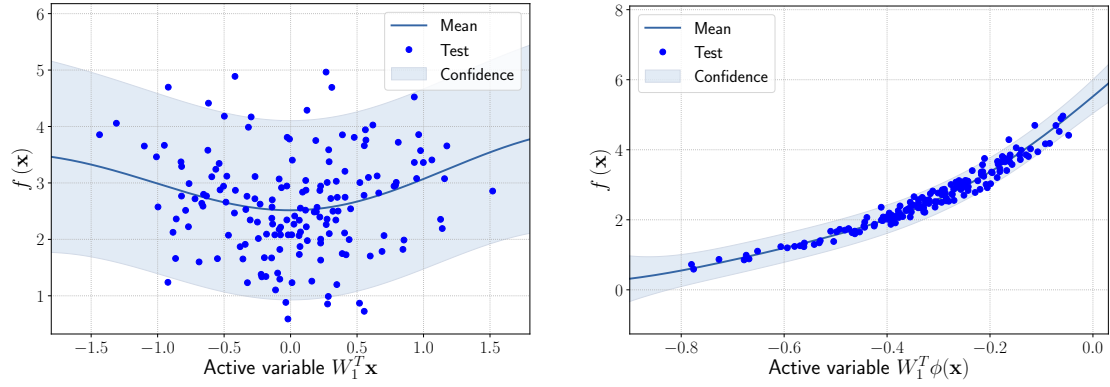


Figure 3: Comparison between the sufficiency summary plots obtained from the application of AS and KAS methods for the hyperparaboloid model function with domain $[-1, 1]^8$, defined in Equation 24. The left plot refers to AS, the right plot to KAS. With the blue solid line we depict the posterior mean of the GP, with the shadow area the 68% confidence intervals, and with the blue dots the testing points.

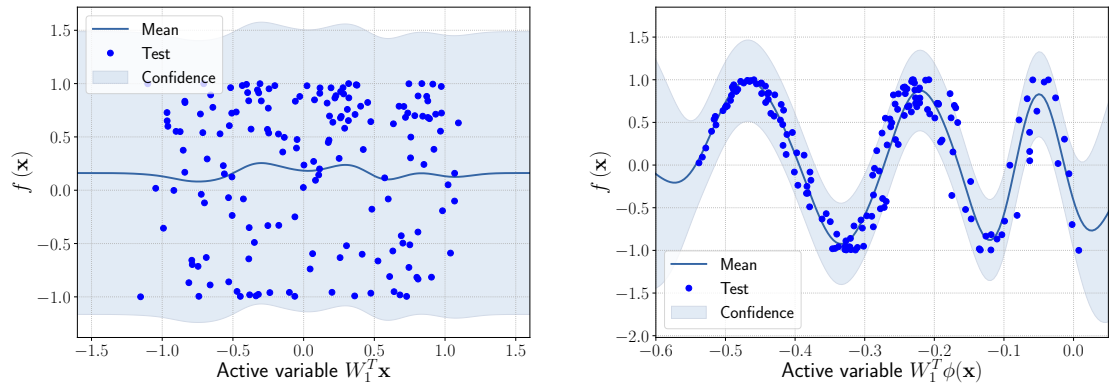


Figure 4: Comparison between the sufficiency summary plots obtained from the application of AS and KAS methods for the surface of revolution model function with domain $[-3, 3]^2$, defined in Equation 25. The left plot refers to AS, the right plot to KAS. With the blue solid line we depict the posterior mean of the GP, with the shadow area the 68% confidence intervals, and with the blue dots the testing points.

4.2 SEIR model for the spread of Ebola

In most engineering applications the output of interest presents a monotonic behaviour with respect to the parameters. This means that, for example, the increment in the inputs produces a proportional response in the outputs. Rarely the model function has a radial symmetry, and in such cases the parameter space can be divided in subdomains, which are analyzed separately. In this section we are going to present a test case where there is no radial symmetry, showing that, even in this case the kernel-based AS presents better performance with respect to AS.

For the Ebola test case³, the output of interest is the basic reproduction number R_0 of the SEIR model, described in [21], which reads

$$R_0 = \frac{\beta_1 + \frac{\beta_2 \rho_1 \gamma_1}{\omega} + \frac{\beta_3}{\gamma_2} \psi}{\gamma_1 + \psi}, \quad (26)$$

with parameters distributed uniformly in $\Omega \subset \mathbb{R}^8$. The parameter space Ω is an hypercube defined by the lower and upper bounds summarized in Table 2.

Table 2: Parameter ranges for the Ebola model. Data taken from [21].

	β_1	β_2	β_3	ρ_1	γ_1	γ_2	ω	ψ
Lower bound	0.1	0.1	0.05	0.41	0.0276	0.081	0.25	0.0833
Upper bound	0.4	0.4	0.2	1	0.1702	0.21	0.5	0.7

We can compare the two one-dimensional response surfaces obtained with Gaussian process regression. The training samples are $N_s = 800$, and we use 1000 features. As spectral measure we use again the multivariate gaussian distribution $\mathcal{N}(\mathbf{0}, \Sigma)$ with hyperparameters the elements of the diagonal of the covariance matrix. The tuning is carried out with 5-fold cross validation. Even in this case, the KAS approach results in smaller RMS with respect to the use of AS (around 60% less), as reported in Table 1. In Figure 5 we report the comparison of the two approaches over an active subspace of dimension 1.

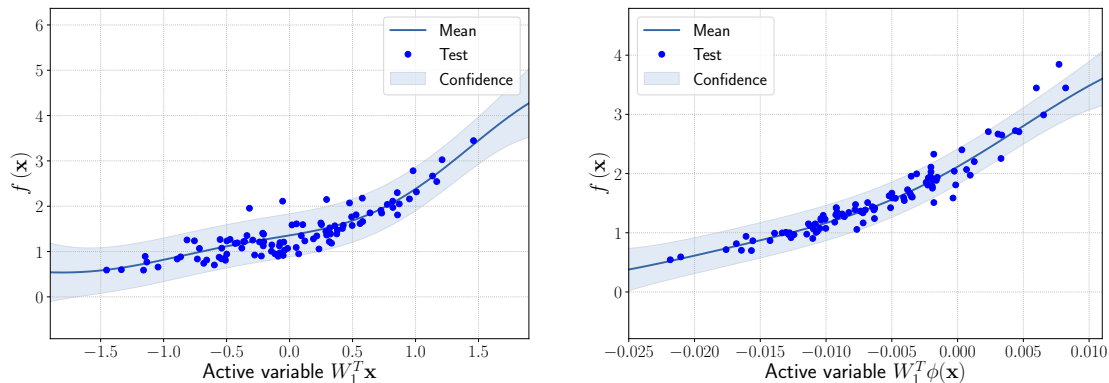


Figure 5: Comparison between the sufficiency summary plots obtained from the application of AS and KAS methods for the R_0 model function with domain Ω , defined in Equation 26. The left plot refers to AS, the right plot to KAS. With the blue solid line we depict the posterior mean of the GP, with the shadow area the 68% confidence intervals, and with the blue dots the testing points.

4.3 Elliptic Partial Differential Equation with random coefficients

In our last benchmark we apply the kernel-based AS to a vectorial model function, that is the solution of a Poisson problem with heterogeneous diffusion coefficient. We refer to [75] for an application, on the same problem, of the AS approach.

³The dataset was taken from <https://github.com/paulcon/as-data-sets>.

We consider the following stochastic Poisson problem on the square $\mathbf{x} = (x, y) \in \Omega := [0, 1]^2$:

$$\begin{cases} -\nabla \cdot (\kappa \nabla u) = 1, & \mathbf{x} \in \Omega, \\ u = 0, & \mathbf{x} \in \partial\Omega_{\text{top}} \cup \partial\Omega_{\text{bottom}}, \\ u = 10y(1 - y), & \mathbf{x} \in \partial\Omega_{\text{left}}, \\ \mathbf{n} \cdot \nabla u = 0, & \mathbf{x} \in \partial\Omega_{\text{right}}, \end{cases} \quad (27)$$

with homogeneous Neumann boundary condition on the right side of the domain, that is $\partial\Omega_{\text{right}}$, Neumann boundary conditions on the left side of the domain, that is $\partial\Omega_{\text{left}}$, and Dirichlet boundary conditions on the remaining part of $\partial\Omega$. The diffusion coefficient $\kappa : (\Omega, \mathcal{A}, P) \times \Omega \rightarrow \mathbb{R}$, where \mathcal{A} is a σ -algebra, is such that $\log(\kappa)$ is a Gaussian random field, with covariance function $C(\mathbf{x}, \mathbf{y})$ defined by

$$C(\mathbf{x}, \mathbf{y}) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{y}\|^2}{\beta^2}\right), \quad \forall \mathbf{x}, \mathbf{y} \in \Omega, \quad (28)$$

where $\beta = 0.03$ is the correlation length. This random field is approximated with the truncated Karhunen-Loève decomposition

$$\kappa(s, \mathbf{x}) \approx \exp\left(\sum_{i=0}^m X_i(s) \gamma_i \psi_i(\mathbf{x})\right) \quad \forall (s, \mathbf{x}) \in \Omega \times \Omega, \quad (29)$$

where $(X_i)_{i \in 1, \dots, m}$ are independent standard normal distributed random variables, and $(\gamma_i, \psi_i)_{i \in 1, \dots, d}$ are the eigenpairs of the Karhunen-Loève decomposition of the zero-mean random field κ .

In our simulation the domain Ω is discretized with a triangular unstructured mesh \mathcal{T} with 3194 triangles. The parameter space has dimension $m = 10$. The simulations are carried out with the finite element method (FEM) with polynomial order one, and for each simulation the parameters $(X_i)_{i=1, \dots, m}$ are sampled from a standard normal distribution. The solution u is evaluated at $d = 1668$ degrees of freedom, thus $(V, R_V) \approx (\mathbb{R}^d, S + M)$ where the metric R_V is approximated with the sum of the stiffness matrix $S \in \mathbb{R}^d \times \mathbb{R}^d$ and the mass matrix $M \in \mathbb{R}^d \times \mathbb{R}^d$. This sum is a discretization of the norm of the Sobolev space $H^1(\Omega)$. The number of features used in the KAS procedure is $D = 1500$, the number of different independent simulations is $M = 1000$.

Three outputs of interest are considered. The first target function $f : \mathbb{R}^m \rightarrow \mathbb{R}$ is the mean value of the solution at the right boundary $\partial\Omega_{\text{right}}$, which reads

$$f(\mathbf{X}) = \frac{1}{|\partial\Omega_{\text{right}}|} \int_{\partial\Omega_{\text{right}}} u(s) ds, \quad (30)$$

and it is used to tune the feature map minimizing the RRMSE of the Gaussian process regression, as described in Algorithm 5. A summary of the results for the first output is reported in Table 1. The plots of the regression are reported in Figure 6. Even in this case both from a qualitative and a quantitative point of view, the kernel-based approach achieves the best results.

The second output we consider is the solution function

$$f : \mathbb{R}^m \rightarrow (V, R_V) \approx (\mathbb{R}^d, S), \quad f(\mathbf{X}) = u \in \mathbb{R}^d. \quad (31)$$

This output can be employed as a surrogate model to predict the solution u given the parameters \mathbf{X} that define the diffusion coefficient instead of carrying out the numerical simulation. The surrogate model should be constructed over the span of the modes identified by the chosen reduction strategy, after projecting the data. AS and KAS modes are distinguished but can detect some common regions of interest as shown in Table 3.

The third output is the evaluation of the solution at a specific degree of freedom with index \hat{i} , that is

$$f : \mathbb{R}^m \rightarrow \mathbb{R}, \quad f(\mathbf{X}) = u_{\hat{i}} \in \mathbb{R}, \quad (32)$$

in this case the dimension of the input space is $m = 100$. Since we use a Lagrangian basis in the finite element formulation and the polynomial order is 1, the node of the mesh associated to the chosen degree of freedom has coordinates $[0.27, 0.427] \in \Omega$. Qualitatively we can see from Table 3 that the AS modes locate features in the domain which are relatively more regular with respect to the KAS modes. To obtain this result we increased the dimension of the input space, otherwise not even the AS modes could locate properly the position in the domain Ω of the degree of freedom.

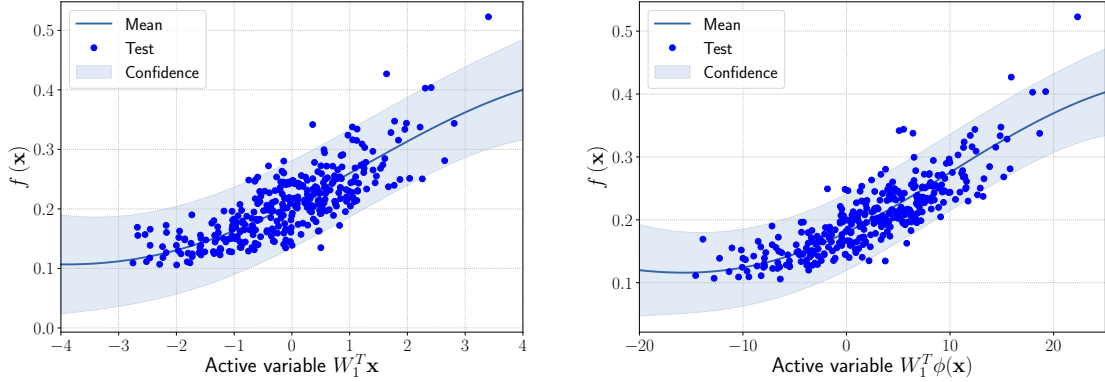


Figure 6: Comparison between the sufficiency summary plots obtained from the application of AS and KAS methods for the stochastic PDE model, defined in Equations (27) and (30). The left plot refers to AS, the right plot to KAS. With the blue solid line we depict the posterior mean of the GP, with the shadow area the 68% confidence intervals, and with the blue dots the testing points.

In the second and third case the diffusion coefficient is given by

$$\kappa(\mathbf{x}) = \exp\left(\sum_{i=1}^D \mathbf{v}_j[i] \tilde{\psi}_j(\mathbf{x})\right) \quad \forall (s, \mathbf{x}) \in \Omega \times \Omega, \quad (33)$$

where $\mathbf{v}_j \in \mathbb{R}^D$, $j \in \{1, \dots, D\}$, is the j -th active eigenvector from the KAS procedure and the functions $\tilde{\Psi} := (\tilde{\psi}_1, \dots, \tilde{\psi}_D)$ are defined by

$$\tilde{\Psi} = \phi(\Psi), \quad (34)$$

where ϕ is the feature map defined in Equation (16) with the projection matrix W and bias b , and $\Psi := (\gamma_1 \psi_1, \dots, \gamma_m \psi_m)$.

The gradients of the three outputs of interest considered are evaluated with the adjoint method.

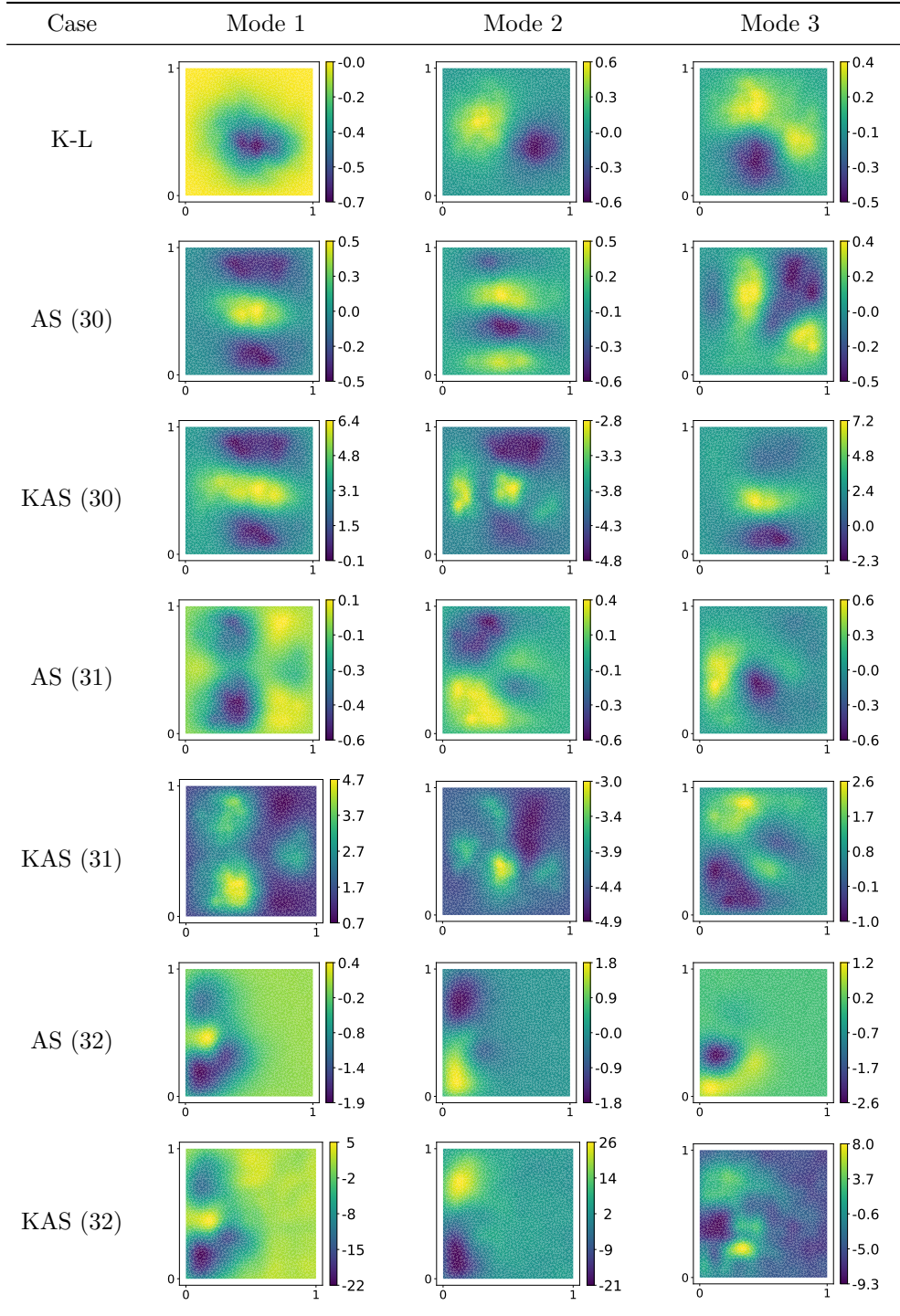
5 A CFD parametric application of KAS solved with the DG method

We want to test the kernel-based extension of the active subspaces in a computational fluid dynamics context. The lift and drag coefficients of a NACA 0012 airfoil are considered as model functions. Numerical simulations are carried out with different input parameters for quantities that describe the geometry and the physical conditions of the problem. The evolution of the model is protracted until a periodic regime is reached. Once the simulation data have been collected, sensitivity analysis is performed searching for an active subspace and response surfaces with GPR are then built from the application of AS and KAS techniques.

The fluid motion is modelled through the unsteady incompressible Navier-Stokes equations approximated through the Chorin-Temam operator-splitting method implemented in HopeFOAM [4]. HopeFOAM is an extension of OpenFOAM [1, 72], an open source software for the solution of complex fluid flows problems, to variable higher order element method and it adopts a Discontinuous Galerkin Method, based on the formulation proposed by Hesthaven and Warburton [28].

The Discontinuous Galerkin method (DG) is a high-order method, which has appealing features such as the low artificial viscosity and a convergence rate which is optimal also on unstructured grids, commonly used in industrial frameworks. In addition to this, DG is naturally suited for the solution of problems described by conservative governing equations (Navier Stokes equations, Maxwell's equations and so on) and for parallel computing. All these properties are due to the fact that, differently from formulations based on standard finite elements, no continuity is imposed on the cell boundaries and neighboring elements only exchange a common flux. The major drawback of DG is its high computational cost with respect to continuous Galerkin methods, due to the

Table 3: First 3 modes using Karhunen-Loève (K-L) decomposition, AS, and KAS, for the outputs defined in Equations (30), (31), and (32).



need of evaluating fluxes during each time step and the presence of extra degree of freedoms in correspondence of the elemental edges.

Nowadays efforts are aimed at applying the DG in problems which involve deformable domains [76] and at improving the computational efficiency of the DG adopting techniques based on hybridization methods, matrix-free implementations, and massive parallelization [42, 45].

5.1 Domain and mesh description

The domain Ω of the fluid dynamic simulation is a two-dimensional duct with a sudden area expansion and a NACA 0012 airfoil is placed in the largest section. The inflow $\partial\Omega_I$ is placed at the beginning of the narrowest part of the duct, and here the fluid velocity is set constant along all the inlet boundary. The outlet is placed on the right hand side and it is denoted with $\partial\Omega_O$. We refer with $\partial\Omega_W := \partial\Omega \setminus \{\partial\Omega_O \cup \partial\Omega_I\}$ to the boundaries of the airfoil and to the walls of the duct, where no slip boundary conditions are applied. The horizontal lengths of the sections of the channels are 0.6 m and 1.35 m, respectively. The vertical length of the duct after the area expansion is 0.4 m, while the width of the first one depends on two distinct parameters. The airfoil has a chord-length equal to 0.1 m but its position with respect to the duct and its angle of attack are described by geometric parameters. Further details about the geometric parameterization of the geometry are provided in the following section. A proper triangulation is designed with the aid of the gmsh [2] tool and the domain is discretized with 4445 unstructured elements.

The evaluation of adimensional magnitudes, commonly used for characterizing the fluid flow field, requires the definition of some reference magnitudes. For the problem at hand we consider the equivalent diameter of the channel in correspondence of the inlet as the reference lengthscale, while the reference velocity is the one imposed at the inlet.

5.2 Parameter space description

We chose 7 heterogeneous parameters for the model: 2 physical, and 5 geometrical which describe the width of the channel and the position of the airfoil. In Table 4 are reported the ranges for the geometrical and physical parameters of the simulation. U is the first component of the initial velocity, ν is the kinematic viscosity, x_0 and y_0 are the horizontal and vertical components of the translation of the airfoil with respect to its reference position (see Figure 7), α is the angle of the counterclockwise rotation and the center of rotation is located right in the middle of the airfoil, y^+ and y^- are the module of the vertical displacements of the upper and lower side of the initial conduct from a prescribed position.

Table 4: Parameter ranges for the NACA problem.

	ν	U	x_0	y_0	α	y^+	y^-
Lower bound	0.00036	0.5	-0.099	-0.035	0	-0.02	-0.02
Upper bound	0.00060	2	0.099	0.035	0.0698	0.02	0.02

In Figure 7 are reported different configurations of the domain for the minimum and maximum values of the parameters α , x_0 , y_0 , and the minimum opening of the channel.

We have considered only the counterclockwise rotation of the airfoil for symmetrical reasons. The range of the Reynolds number varies from 400 to 2000, still under the regime of laminar flow.

5.3 Governing equations

The CFD problem is modeled through the incompressible Navier-Stokes and the open source solver HopeFOAM [4] has been employed for solving this set of equations [28].

Let $\Omega \subset \mathbb{R}^2$ be the two-dimensional domain introduced in subsection 5.1, and let us consider the incompressible Navier-Stokes equations. Omitting the dependence on $(\mathbf{x}, t) \in \Omega \times \mathbb{R}^+$ in the

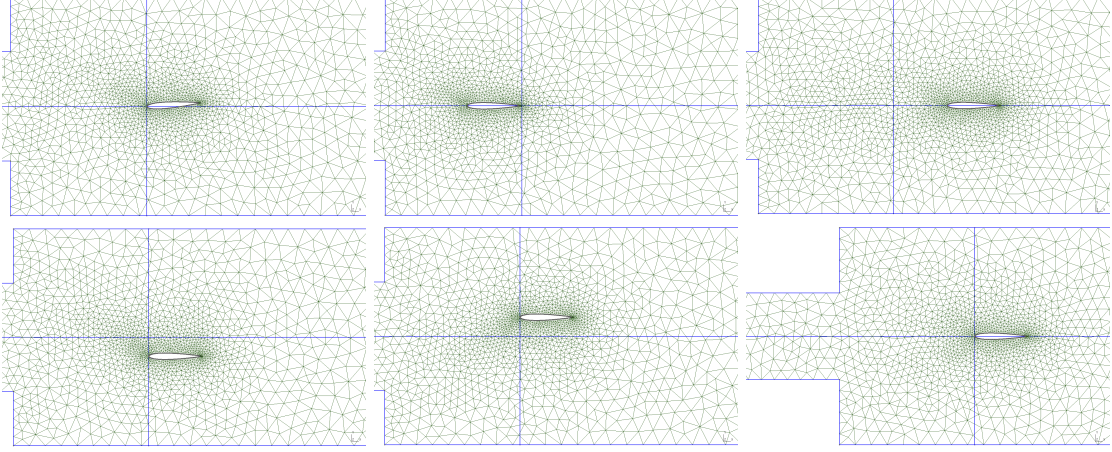


Figure 7: Domain configuration for minimum and maximum values of some geometric parameters. In order are represented the maximum angle of attack α , the ranges for the horizontal translation x_0 , the ranges for the vertical translation y_0 , and the minimum opening of the channel which depends on the parameters y^+ and y^- in Table 4.

first two equations for the sake of compactness, the governing equations are

$$\begin{cases} \partial_t \mathbf{u} + (\mathbf{u} \cdot \nabla) \mathbf{u} = -\nabla p + \nu \Delta \mathbf{u} & \mathbf{x} \in \Omega, \\ \nabla \cdot \mathbf{u} = 0 & \mathbf{x} \in \Omega, \\ \mathbf{u}(\mathbf{x}, 0) = \mathbf{u}_0, \quad p(\mathbf{x}, 0) = 0 & \mathbf{x} \in \Omega, \\ \mathbf{u}(\mathbf{x}, t) = \mathbf{u}_0, \quad \mathbf{n} \cdot \nabla p(\mathbf{x}, t) = 0 & \mathbf{x} \in \partial\Omega_I, \\ \mathbf{u}(\mathbf{x}, t) = 0, \quad \mathbf{n} \cdot \nabla p(\mathbf{x}, t) = 0 & \mathbf{x} \in \partial\Omega_W, \\ \mathbf{n} \cdot \nabla \mathbf{u}(\mathbf{x}, t) = 0, \quad p(\mathbf{x}, t) = 1 & \mathbf{x} \in \partial\Omega_O, \end{cases} \quad (35)$$

where p is the scalar pressure field, $\mathbf{u} = (u, v)$ is the velocity field, ν is the viscosity constant and \mathbf{u}_0 is the initial velocity. In conservative form, the previous equations can be rewritten as

$$\begin{cases} \partial_t \mathbf{u} + \nabla \cdot \mathcal{F} = -\nabla p + \nu \Delta \mathbf{u}, \\ \nabla \cdot \mathbf{u} = 0, \end{cases} \quad (36)$$

with the flux \mathcal{F} given by

$$\mathcal{F} = [\mathbf{F}_1, \mathbf{F}_2] = \begin{bmatrix} u^2 & uv \\ uv & v^2 \end{bmatrix}. \quad (37)$$

From now on, in order to have a more compact notation, the advection term is written as $\mathcal{N}(\mathbf{u}) = \nabla \cdot \mathcal{F}(\mathbf{u})$.

For each timestep the procedure is broken into three stages accordingly to the algorithm proposed by Chorin and adapted for a DG framework by Hesthaven et al. [28]: the solution of the advection dominated conservation law component, the pressure correction weak divergence-free velocity projection, and the viscosity update. The non-linear advection term is treated explicitly in time through a second order Adams-Bashforth method [22], while the diffusion term implicitly. The Chorin algorithm is reported in Algorithm 6.

In order to recover the Discontinuous Galerkin formulation, the equations introduced by the Chorin method are projected onto the solution space by introducing a proper set of test functions and then the variables are approximated over each element as a linear combination of local shape functions. The DG does not impose the continuity of the solution between neighboring elements and therefore it requires the adoption of methods for the evaluation of the flux exchange between neighboring elements. In the present work the convective fluxes are treated accordingly to the Lax-Friedrichs scheme, while the viscous ones are solved through the Interior Penalty method [6, 58].

The aerodynamic quantities we are interested in are the lift and drag coefficients in the incompressible case computed from the quantities \mathbf{u} , p , ν , A_{ref} , and \mathbf{u}_0 with a contour integral along the

Algorithm 6 Chorin Algorithm.

Require: state variables \mathbf{u} and p at $t = 0$, mesh, and boundary conditions

- 1: **while** $t < t_{\text{final}}$ **do**
- 2: Update state variables $\mathbf{u}^{n-1} = \mathbf{u}^n$, $\mathbf{u}^n = \mathbf{u}^{n+1}$.
- 3: Find a guess value for the velocity $\tilde{\mathbf{u}}$ by solving:

$$\frac{\gamma_0 \tilde{\mathbf{u}} - \alpha_0 \mathbf{u}^n - \alpha_1 \mathbf{u}^{n-1}}{\Delta t} = -\beta_0 \mathcal{N}(\mathbf{u}^n) - \beta_1 \mathcal{N}(\mathbf{u}^{n-1}).$$

- 4: Find the pressure at $n + 1$ solving: $-\Delta \bar{p}^{n+1} = -\frac{\gamma_0}{\Delta t} \nabla \cdot \tilde{\mathbf{u}}$.
 - 5: Find the intermediate velocity $\tilde{\tilde{\mathbf{u}}}$ solving: $\gamma_0 \frac{\tilde{\tilde{\mathbf{u}}} - \tilde{\mathbf{u}}}{\Delta t} = \nabla \bar{p}^{n+1}$.
 - 6: Find the velocity at the $n + 1$ time instant solving: $\gamma_0 \left(\frac{\mathbf{u}^{n+1} - \tilde{\tilde{\mathbf{u}}}}{\Delta t} \right) = \nu \Delta \mathbf{u}^{n+1}$.
 - 7: Update t^n .
 - 8: **end while**
 - 9: **return** state variables \mathbf{u} and p at $t = t_{\text{final}}$
-

airfoil Γ as

$$f = \oint_{\Gamma} p \mathbf{n} - \nu (\nabla \mathbf{u} + \nabla \mathbf{u}^T) \mathbf{n} ds. \quad (38)$$

The vector \mathbf{n} is the outward normal along the airfoil surface. The circulation in Γ is affected by both the pressure and stress distributions around the airfoil. The projection of the force along the horizontal and vertical directions gives the drag and lift coefficients respectively

$$C_D = \frac{f \cdot \mathbf{e}_1}{\frac{1}{2} |\mathbf{u}_0|^2 A_{\text{ref}}}, \quad (39)$$

$$C_L = \frac{f \cdot \mathbf{e}_2}{\frac{1}{2} |\mathbf{u}_0|^2 A_{\text{ref}}}, \quad (40)$$

where the reference area A_{ref} is the chord of the airfoil times a length of 1 m. For the aerodynamic analysis of the fluid flow past an airfoil see [33].

5.4 Numerical results

In this section a brief review of the procedure and some details about the numerical method and the computational domain will be presented along the results obtained. For what concerns the DG the polynomial order chosen is 3. The total number of degrees of freedom is 133350. Small variations on the mesh are present in each of the 285 simulations due to the different configurations of the domain. Each simulation is carried out until a periodic behaviour is reached and for this reason the final times range between 3.5 and 5 s, depending on the specific configuration. The integration time intervals are variable and they are updated at the end of each step in order to satisfy the CFL condition. The 7 physical and geometrical parameters of the simulation are sampled uniformly from the intervals in Table 4. In total we consider a dataset of 285 samples.

With the purpose of qualitatively visualizing the results, 4 different simulations are reported in Figure 8 for the module of the velocity field and the scalar pressure field, respectively, both evaluated at the last time instant. These simulations were chosen from the 285 collected in order to show significant differences in the evolution of the fluid flow. In Table 5 are reported the corresponding parameters. Depending on the position of the airfoil and the other physical parameters, different fluid flow patterns can be qualitatively observed.

Table 5: Parameters associated to the simulations plotted in Figure 8.

#	ν	U	x_0	y_0	α	y^+	y^-
1	0.000405	1.99	-0.096	-0.00207	0.00282	0.00784	0.0188
2	0.000541	0.763	-0.084	0.00279	0.0260	-0.0108	0.0195
3	0.000406	0.533	-0.0503	-0.0327	0.0604	-0.0193	0.0068
4	0.000430	1.11	-0.0897	-0.0279	0.0278	-0.00624	0.0197

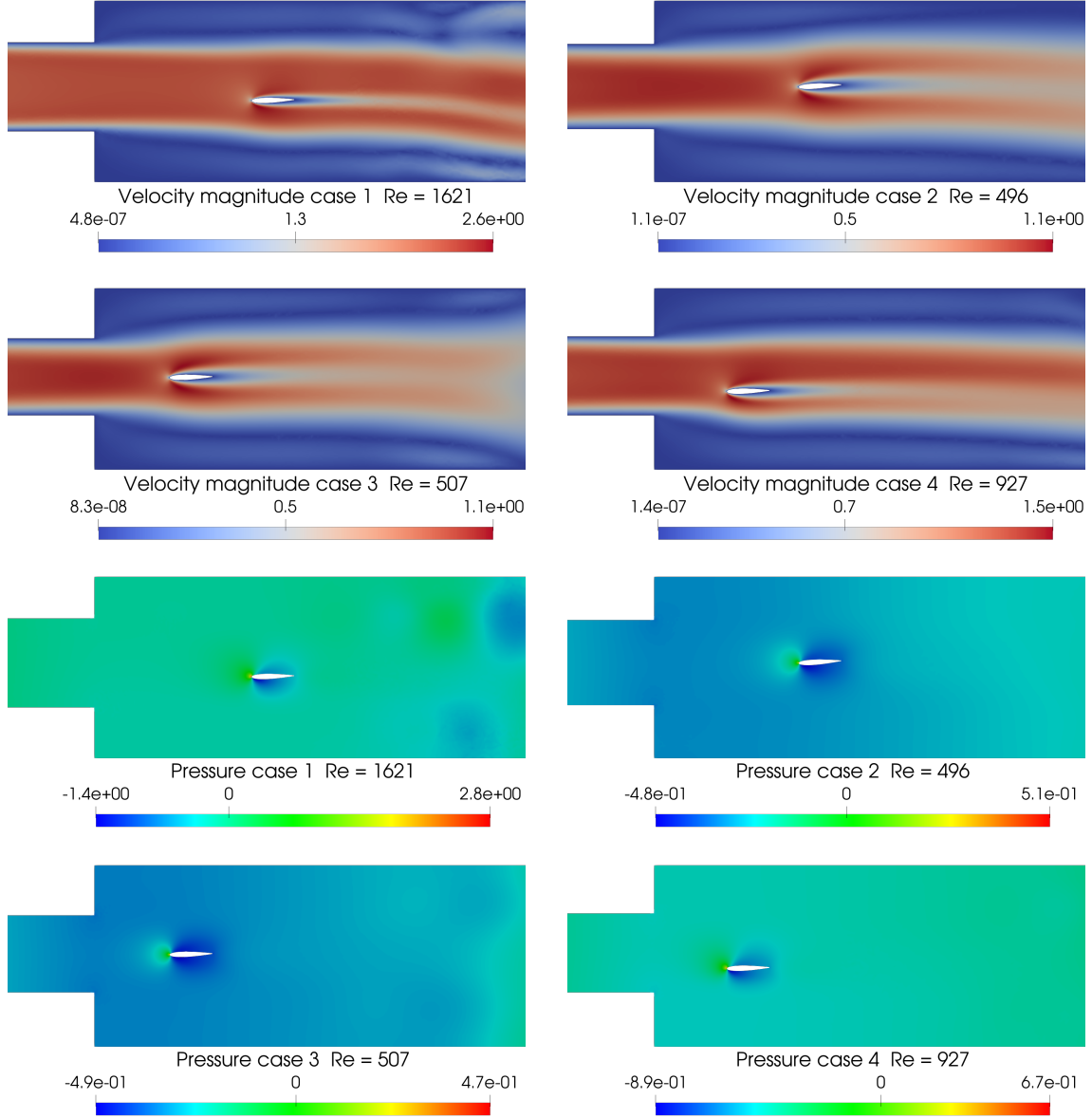


Figure 8: Module of the velocity fields (on the left) and pressure fields (on the right) evaluated at the last time instant of 4 different simulations. The corresponding parameters are reported in Table 5.

The lift (C_L) and drag (C_D) coefficients are evaluated when stationary or periodic regimes are reached, starting from the values of pressure and viscous stresses evaluated on the nodes close to the airfoil. After this sensitivity analysis is carried out. First the AS method is applied. The gradients necessary for the application of the AS method are obtained from the Gaussian process regression of the model functions C_L and C_D on the whole parameters' domain. The eigenvalues of the uncentered covariance matrix for the lift and drag coefficients suggest the presence of a one-dimensional active subspace in both cases.

The plots of the first active eigenvector components are useful as sensitivity measures, see Figure 9. The greater the absolute value of a component is, the greater is its influence on the model function. We observe that the lift coefficient is influenced mainly by the vertical position of the airfoil and the angle of attack, while the drag coefficient depends mainly on the initial velocity, and secondarily on the viscosity and on the angle of attack.

As one could expect from physical considerations, the angle of attack affects both drag and lift coefficients, while the viscosity, which governs the wall stresses, is relevant for the evaluation of the C_D . The vertical position of the airfoil with respect to the symmetric axis of the section of the duct

after the area expansion also greatly affects both coefficients, and this is mainly due to the fact that the fluid flow conditions change drastically between the core, where the speed is higher, and the one close to the wall of the duct, where the speed tends to zero. On the other hand, the horizontal translation has almost no impact on the results, given the regularity of the fluid flow along the x -axis for the considered range of x_0 . Moreover, the non-symmetric behaviour of the upper and lower parameters which determine the opening of the channel is due to the non-symmetric choice of the range considered for the angle of attack.

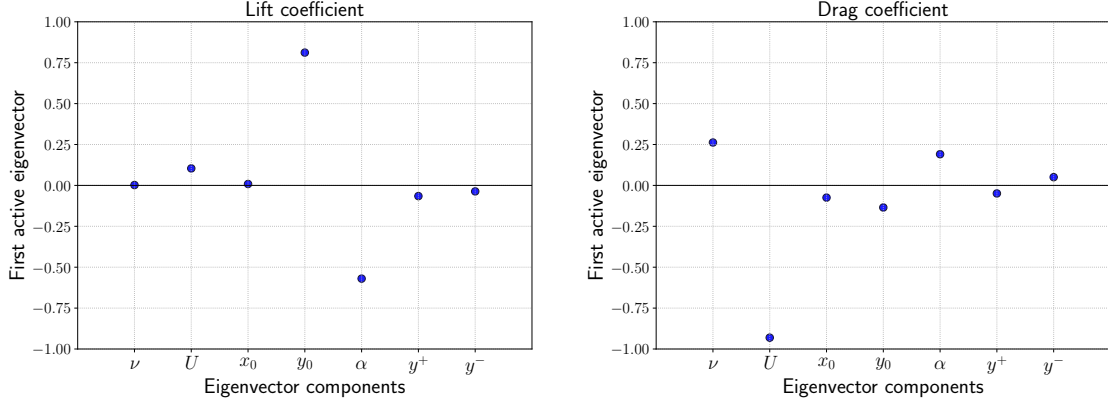


Figure 9: Components of the first active eigenvector for the lift coefficient (on the left), and for the drag coefficient (on the right). Values near 0 suggest little sensitivity for the target function.

The KAS method was applied with 1500 features. In order to compare the AS and KAS methods 5-fold cross validation was implemented. The score of cross validation is the relative root mean square error (RRMSE) defined in Equation 19.

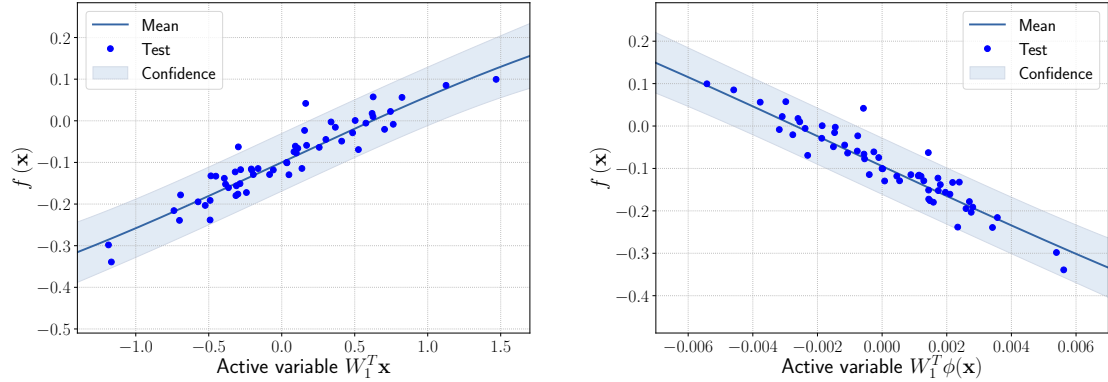


Figure 10: Comparison between the sufficiency summary plots obtained from the application of AS and KAS methods for the lift coefficient C_L defined in Equation 40. The left plot refers to AS, the right plot to KAS. With the blue solid line we depict the posterior mean of the GP, with the shadow area the c68% confidence intervals, and with the blue dots the testing points.

The Gaussian process regressions for the two methods are shown in Figure 10 for the lift coefficient, and in Figure 11 for the drag coefficient. They were obtained as a single step of 5-fold cross validation with one fifth of the 285 samples used as test set. The spectral distribution of the feature map is the Gaussian distribution for the lift, and the Beta for the drag, respectively. The RRMSE mean and standard deviation from 5-fold cross validation, are reported for different active dimensions in Table 6. The feature map from Equation (16) was adopted. The hyperparameters of the spectral distributions were tuned with logarithmic grid-search with 5-fold cross validation as described in Algorithm 5.

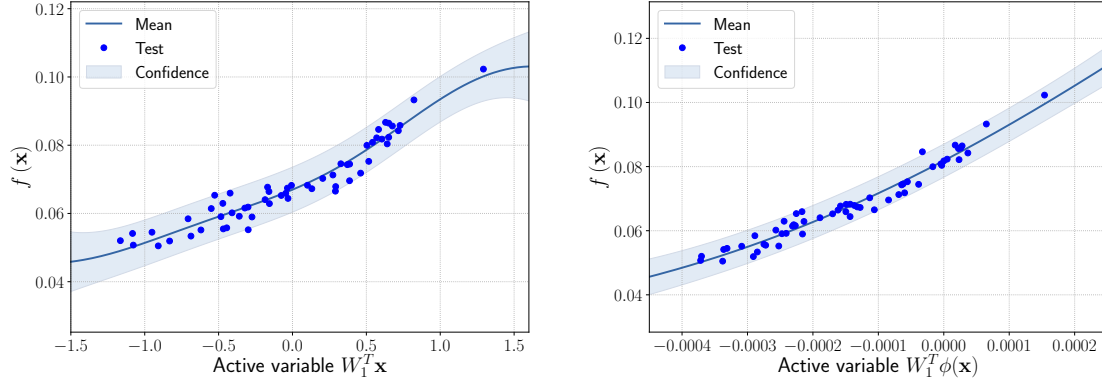


Figure 11: Comparison between the sufficiency summary plots obtained from the application of AS and KAS methods for the drag coefficient C_D defined in Equation 39. The left plot refers to AS, the right plot to KAS. With the blue solid line we depict the posterior mean of the GP, with the shadow area the c68% confidence intervals, and with the blue dots the testing points.

Regarding the drag coefficient, the relative gain using the KAS method reaches the 19.2% on average when employing the Beta spectral measure for the definition of the feature map. The relative gain of the one dimensional response surface built with GPR from the KAS method is 7% on average for the lift coefficient. This result could be due to the higher noise in the evaluation of the C_L . In this case the relative gain increases when the dimension of the response surface increases to 2 with a gain of 14.6%. A slight reduction of the AS RRMSE relative to the drag coefficient is ascertained when increasing the dimension of the response surface.

Table 6: Summary of the results for AS and KAS procedures. In bold the best results.

Method	Dim	Feature space dim	Lift spectral distribution	RRMSE Lift	Drag spectral distribution	RRMSE Drag
AS	1	-	-	0.37 ± 0.09	-	0.268 ± 0.032
KAS	1	1500	$\mathcal{N}(0, \lambda I_d)$	0.344 ± 0.048	Beta(α, β)	0.218 ± 0.045
AS	2	-	-	0.384 ± 0.073	-	0.183 ± 0.027
KAS	2	1500	$\mathcal{N}(0, \lambda I_d)$	0.328 ± 0.071	Beta(α, β)	0.17 ± 0.02

6 Conclusions and perspectives

In this work we presented a new nonlinear extension of the active subspaces property that introduces Kernel-based Active Subspaces (KAS). The method exploits random Fourier features to find active subspaces on high-dimensional feature spaces. We tested the new method over 5 different benchmarks of increasing complexity, and we provided pseudo-codes for every aspects of the proposed kernel-extension. The tested model functions range from scalar to vector-valued. We also provide a CFD application discretized by the Discontinuous Galerkin method. We compared the kernel-based active subspaces to the standard linear active subspaces and we observed in all the cases an increment of the accuracy of the Gaussian response surfaces built over the reduced parameter spaces. The most interesting results regard the possibility to apply the KAS method when an active subspace does not exist. This was shown for radial symmetric model functions.

Future developments will involve the study of more efficient procedures for tuning the hyper-parameters of the spectral distribution. Other possible advances could be done finding an effective back-mapping from the targets to the actual parameters in the full original space. This could promote the implementation of optimization algorithms or other parameter studies enhanced by the kernel-based active subspaces extension.

A Appendix - Proof details

In this section we provide an expanded version of the proof of Theorem 1.

Proof of Theorem 1: existence of an active subspace. The proof is remodeled from [44, 75], and it is developed in five steps:

1. Since $R_V \in \mathcal{M}(d, d)$ is symmetric positive definite there exists a basis of eigenvectors $(\mathbf{w}_i)_{i \in \{1, \dots, d\}}$ and a corresponding set of positive eigenvalues $(\beta_i)_{i \in \{1, \dots, d\}}$ such that

$$R_V = \sum_{i=1}^d \beta_i \mathbf{w}_i \otimes \mathbf{w}_i. \quad (41)$$

2. Let us define the ridge approximation error as

$$\mathbf{e} = \|\mathbf{f} - \mathbf{h} \circ P_r\|_{L^2(\mathbb{R}^m, \mathcal{B}(\mathbb{R}^m), \rho; V)} = \mathbb{E}_P [\|(\mathbf{f}(\mathbf{X}) - \mathbf{h}(P_r(\mathbf{X})))\|_{R_V}^2]. \quad (42)$$

Then we can decompose the error analysis for each component employing the spectral decomposition (1)

$$\begin{aligned} \mathbb{E}_P [\|\mathbf{e}(\mathbf{X})\|_{R_V}^2] &= \mathbb{E}_P [\text{tr}((R_V \mathbf{e}(\mathbf{X})) \otimes \mathbf{e}(\mathbf{X})))] = \\ &= \sum_{i=1}^d \beta_i \mathbb{E}_P [\text{tr}((\mathbf{w}_i \otimes \mathbf{w}_i) \mathbf{e}(\mathbf{X})) \otimes \mathbf{e}(\mathbf{X})] = \\ &= \sum_{i=1}^d \beta_i \mathbb{E}_P [(\mathbf{w}_i \cdot \mathbf{e}(\mathbf{X})) \text{tr}(\mathbf{w}_i \otimes \mathbf{e}(\mathbf{X}))] = \\ &= \sum_{i=1}^d \beta_i \mathbb{E}_P [(\mathbf{w}_i \cdot \mathbf{e}(\mathbf{X}))^2], \end{aligned} \quad (43)$$

so we can define $e_i(\mathbf{X}) = \mathbf{w}_i \cdot \mathbf{e}(\mathbf{X}) = f_i(\mathbf{X}) - h_i(P_r(\mathbf{X}))$, $\forall i \in \{1, \dots, d\}$ and treat each component separately.

3. The next step involves the application of Lemma 1 to the scalar functions $f_i(\mathbf{X}) - h_i(P_r(\mathbf{X}))$, $\forall i \in \{1, \dots, d\}$

$$\begin{aligned} \mathbb{E}_P [(f_i(\mathbf{X}) - h_i(P_r(\mathbf{X})))^2] &= \mathbb{E}_P \left[\mathbb{E}_P [(f_i(\mathbf{X}) - h_i(P_r(\mathbf{X})))^2 | \sigma(P_r)] \right] \\ &\leq \mathbb{E}_P [C_p(\rho, P_r(\mathbf{X})) \mathbb{E}_P [\|(I - P_r^T) \nabla f_i(\mathbf{X})\|_2^2 | \sigma(P_r)]] \\ &\leq \mathbb{E}_P [C_p(\rho, P_r(\mathbf{X}))]^{\frac{1}{p}} \mathbb{E}_P [\|(I - P_r^T) \nabla f_i(\mathbf{X})\|_2^2]^{\frac{1}{q}}, \end{aligned} \quad (44)$$

where we used the Hölder inequality with indexes $(p, q) = (\infty, 1)$ when ρ belongs to the first and second classes of Assumption 3, and $(p, q) = (\frac{\tau+1}{\tau}, 1 + \tau)$ when ρ belongs to the third class.

Then we can bound $\mathbb{E}_P [C_p(\rho, P_r(\mathbf{X}))]^{\frac{1}{p}}$ with a constant $C(C_p(\rho, P_r(\mathbf{X})))$ which depends on the class of ρ (see Lemma 3.1, Lemma 4.2, Lemma 4.3, Lemma 4.4 and Theorem 4.5 of [44]) as follows

$$\begin{aligned} \mathbb{E}_P [C_p(\rho, P_r(\mathbf{X}))]^{\frac{1}{p}} \mathbb{E}_P [\|(I - P_r^T) \nabla f_i(\mathbf{X})\|_2^2]^{\frac{1}{q}} &\leq \\ &\leq C(C_p(\rho, P_r(\mathbf{X}))) \text{tr}(\mathbb{E}_P [(I - P_r^T) \nabla f_i(\mathbf{X}) (\nabla f_i(\mathbf{X}))^T (I - P_r)]^{\frac{1}{q}}) = \\ &= C(C_p(\rho, P_r(\mathbf{X}))) \text{tr}((I - P_r^T) \mathbb{E}_P [\nabla f_i(\mathbf{X}) (\nabla f_i(\mathbf{X}))^T] (I - P_r))^{\frac{1}{q}}. \end{aligned} \quad (45)$$

4. The spectral decomposition (1) is employed again and the covariance matrix H is introduced

in the last equation

$$\begin{aligned}
& \mathbb{E}_P [\|\mathbf{e}(\mathbf{X})\|_{R_V}^2] \leq \\
& \leq \sum_{i=1}^d \beta_i C(C_p(\rho, P_r(\mathbf{X}))) \text{tr}((I - P_r^T) \mathbb{E}_P [((\nabla \mathbf{f}(\mathbf{X}))^T \mathbf{w}_i) \otimes ((\nabla \mathbf{f}(\mathbf{X}))^T \mathbf{w}_i)] (I - P_r))^{\frac{1}{q}} = \\
& = C(C_p(\rho, P_r(\mathbf{X}))) \text{tr}((I - P_r^T) \mathbb{E}_P \left[(\nabla \mathbf{f}(\mathbf{X}))^T \left(\sum_{i=1}^d \beta_i^q \mathbf{w}_i \otimes \mathbf{w}_i \right) \nabla \mathbf{f}(\mathbf{X}) \right] (I - P_r))^{\frac{1}{q}} = \\
& = C(C_p(\rho, P_r(\mathbf{X}))) \text{tr}((I - P_r^T) \mathbb{E}_P [(\nabla \mathbf{f}(\mathbf{X}))^T R_V(\rho) \nabla \mathbf{f}(\mathbf{X})] (I - P_r))^{\frac{1}{q}} = \\
& = C(C_p(\rho, P_r(\mathbf{X}))) \text{tr}((I - P_r^T) H(I - P_r))^{\frac{1}{q}}, \tag{46}
\end{aligned}$$

where $R_V(\rho)$ is the original metric matrix if ρ belongs to the first or second class of Assumption 3 and is equal to

$$\sum_{i=1}^d \beta_i^{1+\tau} \mathbf{w}_i \otimes \mathbf{w}_i, \tag{47}$$

if ρ belongs to the third class.

5. Finally the bound in the statement of the theorem is recovered solving the following minimization problem with classical model reduction arguments employing singular value decomposition (SVD)

$$\tilde{P}_r = \underset{P_r \in \mathcal{O}(m, m)}{\text{argmin}} \text{tr}((I - P_r^T) H(I - P_r)). \tag{48}$$

□

Acknowledgements

This work was partially supported by an industrial Ph.D. grant sponsored by Fincantieri S.p.A. (IRONTH Project), by MIUR (Italian ministry for university and research) through FARE-X-AROMA-CFD project, and partially funded by European Union Funding for Research and Innovation — Horizon 2020 Program — in the framework of European Research Council Executive Agency: H2020 ERC CoG 2015 AROMA-CFD project 681447 “Advanced Reduced Order Methods with Applications in Computational Fluid Dynamics” P.I. Professor Gianluigi Rozza.

References

- [1] <https://openfoam.org/>.
- [2] Gmesh, a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities. <http://gmsh.info/>.
- [3] GPyOpt: A bayesian optimization framework in python. <http://github.com/SheffieldML/GPyOpt>, 2016.
- [4] Hopefoam extension of OpenFOAM. <https://github.com/HopeFOAM/HopeFOAM>, Last release 15th September 2017.
- [5] I. P. Aguiar. Dynamic Active Subspaces: a Data-Driven Approach to Computing Time-Dependent Active Subspaces in Dynamical Systems. Master’s thesis, University of Colorado Boulder, 2018.
- [6] D. N. Arnold. An interior penalty finite element method with discontinuous elements. *SIAM journal on numerical analysis*, 19(4):742–760, 1982. doi:10.1137/0719052.
- [7] E. Barshan, A. Ghodsi, Z. Azimifar, and M. Z. Jahromi. Supervised principal component analysis: Visualization, classification and regression on subspaces and submanifolds. *Pattern Recognition*, 44(7):1357–1371, 2011.

- [8] A. Berlinet and C. Thomas-Agnan. *Reproducing kernel Hilbert spaces in probability and statistics*. Springer Science & Business Media, 2011.
- [9] A. Bobrowski. *Functional analysis for probability and stochastic processes: an introduction*. Cambridge University Press, 2005.
- [10] R. A. Bridges, A. D. Gruber, C. Felder, M. E. Verma, and C. Hoff. Active Manifolds: A nonlinear analogue to Active Subspaces. In *Proceedings of International Conference on Machine Learning*, 2019.
- [11] S. L. Brunton and J. N. Kutz. *Data-driven science and engineering: Machine learning, dynamical systems, and control*. Cambridge University Press, 2019.
- [12] P. G. Constantine. *Active subspaces: Emerging ideas for dimension reduction in parameter studies*, volume 2 of *SIAM Spotlights*. SIAM, 2015.
- [13] P. G. Constantine and P. Diaz. Global sensitivity metrics from active subspaces. *Reliability Engineering & System Safety*, 162:1–13, 2017.
- [14] P. G. Constantine, E. Dow, and Q. Wang. Active subspace methods in theory and practice: applications to kriging surfaces. *SIAM Journal on Scientific Computing*, 36(4):A1500–A1524, 2014.
- [15] P. G. Constantine, M. Emory, J. Larsson, and G. Iaccarino. Exploiting active subspaces to quantify uncertainty in the numerical simulation of the HyShot II scramjet. *Journal of Computational Physics*, 302:1–20, 2015.
- [16] R. D. Cook and L. Ni. Sufficient dimension reduction via inverse regression: A minimum discrepancy approach. *Journal of the American Statistical Association*, 100(470):410–428, 2005.
- [17] C. Cui, K. Zhang, T. Daulbaev, J. Gusak, I. Oseledets, and Z. Zhang. Active subspace of neural networks: Structural analysis and universal attacks. *SIAM Journal on Mathematics of Data Science*, 2(4):1096–1122, 2020. doi:10.1137/19M1296070.
- [18] N. Demo, M. Tezzele, A. Mola, and G. Rozza. Hull Shape Design Optimization with Parameter Space and Model Reductions, and Self-Learning Mesh Morphing. *Journal of Marine Science and Engineering*, 9(2):185, 2021. doi:10.3390/jmse9020185.
- [19] N. Demo, M. Tezzele, and G. Rozza. A non-intrusive approach for reconstruction of POD modal coefficients through active subspaces. *Comptes Rendus Mécanique de l’Académie des Sciences, DataBEST 2019 Special Issue*, 347(11):873–881, November 2019. doi:10.1016/j.crme.2019.11.012.
- [20] N. Demo, M. Tezzele, and G. Rozza. A Supervised Learning Approach Involving Active Subspaces for an Efficient Genetic Algorithm in High-Dimensional Optimization Problems. *SIAM Journal on Scientific Computing*, 43(3):B831–B853, 2021. doi:10.1137/20M1345219.
- [21] P. Diaz, P. Constantine, K. Kalmbach, E. Jones, and S. Pankavich. A modified SEIR model for the spread of Ebola in Western Africa and metrics for resource allocation. *Applied Mathematics and Computation*, 324:141–155, 2018.
- [22] J. Gazdag. Time-differencing schemes and transform methods. *Journal of Computational Physics*, 20(2):196–207, 1976.
- [23] S. F. Ghoreishi, S. Friedman, and D. L. Allaire. Adaptive Dimensionality Reduction for Fast Sequential Optimization With Gaussian Processes. *Journal of Mechanical Design*, 141(7):071404, 2019.
- [24] GPy. GPy: A Gaussian process framework in Python. <http://github.com/SheffieldML/GPy>, since 2012.
- [25] M. Guo and J. S. Hesthaven. Reduced order modeling for nonlinear structural analysis using Gaussian process regression. *Computer Methods in Applied Mechanics and Engineering*, 341:807–826, 2018.

- [26] P. Héas, C. Herzet, and B. Combes. Generalized kernel-based dynamic mode decomposition. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3877–3881. IEEE, 2020.
- [27] J. S. Hesthaven, G. Rozza, and B. Stamm. *Certified Reduced Basis Methods for Parametrized Partial Differential Equations*. Springer, 2016.
- [28] J. S. Hesthaven and T. Warburton. *Nodal discontinuous Galerkin methods: algorithms, analysis, and applications*. Springer Science & Business Media, 2007.
- [29] J. L. Jefferson, J. M. Gilbert, P. G. Constantine, and R. M. Maxwell. Active subspaces for sensitivity analysis and dimension reduction of an integrated hydrologic model. *Computers & geosciences*, 83:127–138, 2015.
- [30] W. Ji, Z. Ren, Y. Marzouk, and C. K. Law. Quantifying kinetic uncertainty in turbulent combustion simulations using active subspaces. *Proceedings of the Combustion Institute*, 37(2):2175–2182, 2019. doi:10.1016/j.proci.2018.06.206.
- [31] W. Ji, J. Wang, O. Zahm, Y. M. Marzouk, B. Yang, Z. Ren, and C. K. Law. Shared low-dimensional subspaces for propagating kinetic uncertainty to multiple outputs. *Combustion and Flame*, 190:146–157, 2018.
- [32] I. Kevrekidis, C. Rowley, and M. Williams. A kernel-based method for data-driven Koopman spectral analysis. *Journal of Computational Dynamics*, 2(2):247–265, 2015.
- [33] P. K. Kundu, I. M. Cohen, and D. R. Dowling, editors. *Fluid Mechanics (Fifth Edition)*. Academic Press, Boston, 2012. doi:https://doi.org/10.1016/C2009-0-63410-3.
- [34] K.-C. Li. Sliced inverse regression for dimension reduction. *Journal of the American Statistical Association*, 86(414):316–327, 1991.
- [35] L. Li. Sparse sufficient dimension reduction. *Biometrika*, 94(3):603–613, 2007.
- [36] Z. Li, J.-F. Ton, D. Oglic, and D. Sejdinovic. Towards a unified analysis of random Fourier features. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 3905–3914. ICML, 2019.
- [37] T. W. Lukaczyk. *Surrogate Modeling and Active Subspaces for Efficient Optimization of Supersonic Aircraft*. PhD thesis, Stanford University, 2015.
- [38] T. W. Lukaczyk, P. Constantine, F. Palacios, and J. J. Alonso. Active subspaces for shape optimization. In *10th AIAA multidisciplinary design optimization conference*, page 1171, 2014.
- [39] L. Meneghetti, N. Demo, and G. Rozza. A dimensionality reduction approach for convolutional neural networks. *arXiv preprint arXiv:2110.09163*, Submitted, 2021.
- [40] S. Mika, B. Schölkopf, A. J. Smola, K.-R. Müller, M. Scholz, and G. Rätsch. Kernel PCA and de-noising in feature spaces. In *Advances in neural information processing systems*, pages 536–542, 1999.
- [41] M. Mohri, A. Rostamizadeh, and A. Talwalkar. *Foundations of machine learning*. MIT press, 2018.
- [42] N. C. Nguyen, J. Peraire, and B. Cockburn. An implicit high-order hybridizable discontinuous galerkin method for linear convection–diffusion equations. *Journal of Computational Physics*, 228(9):3232–3254, 2009.
- [43] M. Palaci-Olgun. Gaussian process modeling and supervised dimensionality reduction algorithms via stiefel manifold learning. Master’s thesis, University of Toronto Institute of Aerospace Studies, 2018.
- [44] M. T. Parente, J. Wallin, B. Wohlmuth, et al. Generalized bounds for active subspaces. *Electronic Journal of Statistics*, 14(1):917–943, 2020.

- [45] W. Pazner and P.-O. Persson. Stage-parallel fully implicit Runge–Kutta solvers for discontinuous Galerkin fluid simulations. *Journal of Computational Physics*, 335:700–717, 2017. doi:10.1016/j.jcp.2017.01.050.
- [46] A. Quarteroni and G. Rozza. *Reduced order methods for modeling and computational reduction*, volume 9. Springer, 2014. doi:10.1007/978-3-319-02090-7.
- [47] A. Rahimi and B. Recht. Random features for large-scale kernel machines. In *Advances in neural information processing systems*, pages 1177–1184, 2008.
- [48] F. Romor, M. Tezzele, M. Mrosek, C. Othmer, and G. Rozza. Multi-fidelity data fusion through parameter space reduction with applications to automotive engineering. *arXiv preprint arXiv:2110.14396*, Submitted, 2021.
- [49] F. Romor, M. Tezzele, and G. Rozza. ATHENA: Advanced Techniques for High dimensional parameter spaces to Enhance Numerical Analysis. *Software Impacts*, 10:100133, 2021. doi:10.1016/j.simpa.2021.100133.
- [50] F. Romor, M. Tezzele, and G. Rozza. A local approach to parameter space reduction for regression and classification tasks. *arXiv preprint arXiv:2107.10867*, Submitted, 2021.
- [51] G. Rozza, M. Hess, G. Stabile, M. Tezzele, and F. Ballarin. Basic Ideas and Tools for Projection-Based Model Reduction of Parametric Partial Differential Equations. In P. Benner, S. Grivet-Talocia, A. Quarteroni, G. Rozza, W. H. A. Schilders, and L. M. Silveira, editors, *Model Order Reduction*, volume 2, chapter 1, pages 1–47. De Gruyter, Berlin, Boston, 2020. doi:10.1515/9783110671490-001.
- [52] G. Rozza, M. H. Malik, N. Demo, M. Tezzele, M. Girfoglio, G. Stabile, and A. Mola. Advances in Reduced Order Methods for Parametric Industrial Problems in Computational Fluid Dynamics. In R. Owen, R. de Borst, J. Reese, and P. Chris, editors, *ECCOMAS ECFD 7 - Proceedings of 6th European Conference on Computational Mechanics (ECCM 6) and 7th European Conference on Computational Fluid Dynamics (ECFD 7)*, pages 59–76, Glasgow, UK, 2018.
- [53] T. M. Russi. *Uncertainty quantification with experimental data and complex system models*. PhD thesis, UC Berkeley, 2010.
- [54] F. Salmoiraghi, F. Ballarin, G. Corsi, A. Mola, M. Tezzele, and G. Rozza. Advances in geometrical parametrization and reduced order models and methods for computational fluid dynamics problems in applied sciences and engineering: Overview and perspectives. *ECCOMAS Congress 2016 - Proceedings of the 7th European Congress on Computational Methods in Applied Sciences and Engineering*, 1:1013–1031, 2016. doi:10.7712/100016.1867.8680.
- [55] B. Schölkopf, A. Smola, and K.-R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural computation*, 10(5):1299–1319, 1998.
- [56] B. Schölkopf, A. J. Smola, F. Bach, et al. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2002.
- [57] P. Seshadri, S. Shahpar, P. Constantine, G. Parks, and M. Adams. Turbomachinery active subspace performance maps. *Journal of Turbomachinery*, 140(4):041003, 2018. doi:10.1115/1.4038839.
- [58] K. Shahbazi. An explicit expression for the penalty parameter of the interior penalty method. *Journal of Computational Physics*, 205(2):401–407, 2005. doi:10.1016/j.jcp.2004.11.017.
- [59] B. Sriperumbudur and N. Sterge. Approximate kernel PCA using random features: computational vs. statistical trade-off. *arXiv preprint arXiv:1706.06296*, 2017.
- [60] M. Tezzele, F. Ballarin, and G. Rozza. Combined parameter and model reduction of cardiovascular problems by means of active subspaces and POD-Galerkin methods. In D. Boffi, L. F. Pavarino, G. Rozza, S. Scacchi, and C. Vergara, editors, *Mathematical and Numerical Modeling of the Cardiovascular System and Applications*, volume 16 of *SEMA-SIMAI Series*, pages 185–207. Springer International Publishing, 2018. doi:10.1007/978-3-319-96649-6_8.

- [61] M. Tezzele, N. Demo, M. Gadalla, A. Mola, and G. Rozza. Model order reduction by means of active subspaces and dynamic mode decomposition for parametric hull shape design hydrodynamics. In *Technology and Science for the Ships of the Future: Proceedings of NAV 2018: 19th International Conference on Ship & Maritime Research*, pages 569–576. IOS Press, 2018. doi:10.3233/978-1-61499-870-9-569.
- [62] M. Tezzele, N. Demo, A. Mola, and G. Rozza. An integrated data-driven computational pipeline with model order reduction for industrial and applied mathematics. In M. Günther and W. Schilders, editors, *Novel Mathematics Inspired by Industrial Challenges*, number 38 in Mathematics in Industry. Springer International Publishing, 2022. doi:10.1007/978-3-030-96173-2_7.
- [63] M. Tezzele, N. Demo, and G. Rozza. Shape optimization through proper orthogonal decomposition with interpolation and dynamic mode decomposition enhanced by active subspaces. In R. Bensow and J. Ringsberg, editors, *Proceedings of MARINE 2019: VIII International Conference on Computational Methods in Marine Engineering*, pages 122–133, 2019.
- [64] M. Tezzele, N. Demo, G. Stabile, A. Mola, and G. Rozza. Enhancing CFD predictions in shape design problems by model and parameter space reduction. *Advanced Modeling and Simulation in Engineering Sciences*, 7(40), 2020. doi:10.1186/s40323-020-00177-y.
- [65] M. Tezzele, L. Fabris, M. Sidari, M. Sicchiero, and G. Rozza. A multi-fidelity approach coupling parameter space reduction and non-intrusive POD with application to structural optimization of passenger ship hulls. *arXiv preprint arXiv:2206.01243*, Submitted, 2022.
- [66] M. Tezzele, F. Salmoiraghi, A. Mola, and G. Rozza. Dimension reduction in heterogeneous parametric spaces with application to naval engineering shape design problems. *Advanced Modeling and Simulation in Engineering Sciences*, 5(1):25, Sep 2018. doi:10.1186/s40323-018-0118-3.
- [67] R. Tripathy and I. Bilonis. Deep active subspaces: A scalable method for high-dimensional uncertainty propagation. In *ASME 2019 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. American Society of Mechanical Engineers Digital Collection, 2019.
- [68] R. Tripathy, I. Bilonis, and M. Gonzalez. Gaussian processes with built-in dimensionality reduction: Applications to high-dimensional uncertainty propagation. *Journal of Computational Physics*, 321:191–223, 2016. doi:10.1016/j.jcp.2016.05.039.
- [69] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. Jarrod Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. Carey, Í. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and S. . . Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 2020. doi:https://doi.org/10.1038/s41592-019-0686-2.
- [70] M. Vohra, A. Alexanderian, H. Guy, and S. Mahadevan. Active subspace-based dimension reduction for chemical kinetics applications with epistemic uncertainty. *Combustion and Flame*, 204:152–161, 2019. doi:10.1016/j.combustflame.2019.03.006.
- [71] F. W. Warner. *Foundations of differentiable manifolds and Lie groups*, volume 94. Springer Science & Business Media, 1983.
- [72] H. G. Weller, G. Tabor, H. Jasak, and C. Fureby. A tensorial approach to computational continuum mechanics using object-oriented techniques. *Computers in physics*, 12(6):620–631, 1998. doi:10.1063/1.168744.
- [73] C. K. Williams and C. E. Rasmussen. *Gaussian Processes for Machine Learning*. Adaptive Computation and Machine Learning series. MIT press Cambridge, MA, 2006.
- [74] Q. Wu, S. Mukherjee, and F. Liang. Localized sliced inverse regression. In *Advances in neural information processing systems*, pages 1785–1792, 2009.

- [75] O. Zahm, P. G. Constantine, C. Prieur, and Y. M. Marzouk. Gradient-based dimension reduction of multivariate vector-valued functions. *SIAM Journal on Scientific Computing*, 42(1):A534–A558, 2020. doi:10.1137/18M1221837.
- [76] M. J. Zahr and P.-O. Persson. An adjoint method for a high-order discretization of deforming domain conservation laws for optimization of flow problems. *Journal of Computational Physics*, 326:516–543, 2016.
- [77] G. Zhang, J. Zhang, and J. Hinkle. Learning nonlinear level sets for dimensionality reduction in function approximation. In *Advances in Neural Information Processing Systems*, pages 13199–13208, 2019.