Journal of Computational Science 85 (2025) 102511

Contents lists available at ScienceDirect

Journal of Computational Science

journal homepage: www.elsevier.com/locate/jocs

Integration of multi-fidelity methods in parametrized non-intrusive reduced order models for industrial applications

Fausto Dicech [a,b,*], Konstantinos Gkaragkounis [c], Lucia Parussini [a], Anna Spagnolo [b], Haysam Telib [c]

[a] Department of Engineering and Architecture, Università degli Studi di Trieste, Via Valerio 6/1, Trieste, 34127, Italy
[b] Development Centre, Optimad S.r.l, Località Padriciano, 99, Trieste, 34149, Italy
[c] Head Office, Optimad S.r.l, Via Agostino da Montefeltro, 2, Torino, 10122, Italy

ARTICLE INFO

Keywords:
Reduced order models
Multi-fidelity
Road vehicles
Proper Orthogonal Decomposition
Aerodynamics

ABSTRACT

Exploring the behavior of complex industrial problems might become burdensome, especially in high-dimensional design spaces. Reduced Order Models (ROMs) aim to minimize the computational effort needed to study different design choices by exploiting already available data. In this work, we propose a methodology where the full-order solution is replaced with a Proper Orthogonal Decomposition based ROM, enhanced by a multi-fidelity surrogate model. Multi-fidelity approaches allow to exploit heterogeneous information sources, and consequently reduce the cost of creating the training data needed to build the ROM. To explore the multi-fidelity ROM capabilities, we present and discuss results and challenges for an automotive aerodynamic application, based on a geometric morphing of the DrivAer test case with multi-fidelity fluid-dynamics simulations.

1. Introduction

Nowadays, the necessity to explore bigger and wider parameter spaces is growing, especially in industrial design problems, such as external aerodynamics. The optimizations increase in complexity and so does the need for considerable amounts of Computational Fluid Dynamics (CFD) simulations. When dealing with High-fidelity (HF) simulations and many parameters, the complexity tends to be intractable. To reduce this burden, approximation techniques have been developed, such as Reduced Order Models (ROMs). In recent years ROMs have been applied in different fields, from CFD [1,2] to experimental fluid-dynamics [3,4], structural analysis, and many other fields [5–7]. A ROM aims to approximate full-order solutions of a given problem, usually discretized on mesh, or on a finite set of points in space. Consequently, ROMs demonstrate proficiency in approximating diffused quantities, such as velocity and pressure fields, when applied in the context of fluid dynamics. In particular, Proper Orthogonal Decomposition (POD)-based ones showed great results so far [8,9]. With POD ROMs it is possible to approximate the full-order CFD solution. Once trained, these models are characterized by a remarkable accuracy, robustness and almost real-time predictions.

Nevertheless, a certain amount of offline HF simulations are still required to train the POD ROM. This has a significant impact on the overall cost of the model. In this work, we aim to exploit multi-fidelity methods to lower the number of HF simulations. Multi-fidelity models leverage Low-fidelity (LF) data and their relationship with HF information in order to improve our knowledge of the HF problem. To give a better understanding of the methodology, especially regarding its potentialities and technical complexities, we present alongside it an application for an industrial problem.

Most research on multi-fidelity regression models has focused on the development of multi-fidelity surrogates of a single quantity of interest. There are multi-fidelity kriging approaches, where the key point is the covariance matrix between LF and HF model [10]. In [11,12] the cokriging algorithm of [10] was reformulated in a recursive manner to reduce computational complexity.

Alternative approaches for the creation of the covariance matrix between LF and HF models are proposed in hierarchical kriging models [13,14] and the Non-linear AutoRegressive multi-fidelity Gaussian

Abbreviations: CAD, Computer Aided Design; C_D, Coefficient of Drag; CFD, Computational Fluid Dynamics; DL, Deep Learning; DoE, Design of Experiment; GP, Gaussian Process; HF, High-fidelity; LF, Low-fidelity; LHS, Latin Hypercube Sampling; MLE, Maximum Likelihood Estimation; NARGP, Non-linear AutoRegressive multi-fidelity Gaussian Process regression; NI, Non-Intrusive; NN, Neural Networks; PDE, Partial Differential Equation; POD, Proper Orthogonal Decomposition; RBF, Radial Basis Function; ROM, Reduced Order Model; SVD, Singular Value Decomposition

* Department of Engineering and Architecture, Università degli Studi di Trieste, Via Valerio 6/1, Trieste, 34127, Italy.
E-mail address: fausto.dicech@phd.units.it (F. Dicech).

https://doi.org/10.1016/j.jocs.2024.102511
Received 27 August 2024; Received in revised form 10 December 2024; Accepted 18 December 2024
Available online 6 January 2025

Process regression (NARGP) [15]. There are correction-based methods, where a bridge function or scaling function models the differences between HF and LF models [16–18]. More recently, multi-fidelity stochastic radial basis functions metamodel [19], multi-fidelity NN models [20,21], multi-fidelity deep NN models [22–24], multi-fidelity deep operator networks [25,26], and multi-fidelity Physics Informed Deep NN with transfer learning [27,28] have been explored.

The adoption of multi-fidelity approaches for surrogate models of parametrized fields is a fairly new trend, as shown below by some techniques found in the literature.

In [29] the main idea is to use a POD ROM to encode the difference between the field solution obtained on two different grids, a coarse one and a fine one. An approximation of the fine grid solution is obtained summing a coarse grid solution and the POD approximation. Similarly, [30,31] proposed Gappy POD and constrained Gappy POD method, respectively. In [32,33] a multi-fidelity non intrusive POD methodology is presented, appending the POD basis with LF modes projected on the complementary space spanned by the HF data available. In [34] a multi-fidelity surrogate model using POD is proposed, where HF and LF data are combined to build the snapshot matrix, and trained the ROM with a multi-fidelity RBF. This methodology does not take into account differences in the meshes, consequential to the geometrical parameterization, or different refinements between LF and HF meshes. Nonetheless, in [7,35,36] a manifold alignment method is presented to fuse inconsistent fields from HF and LF simulations by individually projecting their solution onto a common shared latent space. To train the ROM, different models have been employed, such as hierarchical kriging, and multi-fidelity RBF.

In [37–40] kriging meta-models to approximate the POD coefficients are extended to a multi-fidelity context. A non-intrusive (NI) reduced basis method for parametrized nonlinear partial differential equations was introduced in [41], where multi-fidelity Gaussian process regression is employed to approximate the combination coefficients of the reduced basis. In [42] an artificial neural network mapped the relationship between the POD coefficients from different fidelity solutions. In [43] a comparison of utilizing kriging and artificial neural network to map this relationship can be found.

Furthermore, [44] proposed a multi-fidelity multi-step Neural Network (NN) model, [45] a multi-fidelity multi-step Bayesian NN, [46] multi-fidelity concatenated NN, [47] multi-fidelity convolutional autoencoders, [48] multi-fidelity deep NNs for Bayesian model updating, [49,50] transfer learning for field reconstruction based on multi-fidelity solutions. Other methods have different approaches to the multi-fidelity problem, such as [51]. Here, a deep-learning (DL) - based POD ROM is presented and exploits multi-fidelity information in the pretraining stage of the DL model, thereby reducing offline computational effort.

Recently, [52] has proposed a multi-fidelity surrogate based on Galerkin-POD model, interpreting closure problem to compensate for the contribution of the truncated scales onto the resolved ones as a multi-fidelity problem and using a multi-fidelity deep operator network (DeepONet) framework to address it.

With this in mind, the proposed POD multi-fidelity ROM wants to address sparse HF problems, eventually characterized by large geometrical deformations, which implies non-coherent mesh discretizations and requires a mapping strategy. In addition, we opted to use the NARGP [15] unlike the other prevalent multi-fidelity models, resulting in an enhanced level of the ROM's flexibility. Furthermore, mixing different fidelity snapshots in the decomposition of the snapshot space has been investigated. This approach has been developed in view of tackling industrial problems, i.e. external aerodynamics, where POD ROMs might be highly beneficial, especially for topology optimization and simulating fluid–structure interactions in multi-physics simulations [53], or any application requiring multiple queries from a CFD solver.

Here, the performance of the proposed POD multi-fidelity ROM has been quantified on an automotive-based benchmark for external aerodynamics problems, namely the DrivAer [54,55], for the pressure distribution prediction when having geometrical deformations. Through this application, it is possible to comprehend how this multi-fidelity ROM addresses challenges inherent to industrial design processes for external aerodynamics, such as wide-ranging geometries and heterogeneous CFD simulation results.

The paper is structured as follows. In Section 2 the methodology is presented. Successively, the results are shown in Section 3, where the test case is illustrated in 3.1, the model set up in 3.2, and then the results achieved in 3.3. Finally, in Section 4 the previous results are discussed and the conclusions given.

2. Method

Below, in Sections 2.1 and 2.2 the fundamental components of the model are introduced, the POD-based ROMs and multi-fidelity models respectively. The actual POD multi-fidelity ROM is detailed in Section 2.3, with emphasis on the process and the mapping of non-coherent snapshots.

2.1. POD-based reduced order models

In order to design a ROM, a dimensionality reduction is often required. One of the most common alternatives is the POD [56], also known as Karhunen–Loeve Expansion. In this work we will employ the POD, as in Section 2.1.1.

Subsequently, ROMs can be classified into intrusive ROMs and NI ROMs. When the dimensionality reduction technique has access to the problem governing equation, usually a set of discretized Partial Differential Equations (PDEs), we refer to it as intrusive ROM. A typical example is Galerkin or Petrov–Galerkin projection ROMs. Access to the source code for the simulation is a fundamental requirement for intrusive models. By doing so, intrusive ROMs are strongly related to the physical description of the problem. However, this is not always a viable option. On the other hand, NI ROMs are purely data-driven since they require only the fields resulting from a CFD simulation or any other source, independently of the used tool.

NI ROMs are based on machine learning algorithms, where the most common are RBF interpolation, GP regression, or NN. By combining these regression models with the encoded representation obtained with the POD, or other dimension reduction techniques, a ROM can be trained. The proposed POD multi-fidelity ROM will be part of the NI ROM class.

2.1.1. Proper orthogonal decomposition

POD describes a high-dimensional dataset through a set of orthogonal basis functions [56,57]. Formally, these basis functions are obtained as the solution of an optimization problem to capture most of the energy norm associated with the basis.

Let $\mathbf{S} = \{\mathbf{u}_1, \dots, \mathbf{u}_n\}$ be a set of vectors, each one representing a target field $\mathbf{u}_i \in \mathbb{R}^N \, \forall \, i \in [1, \dots, n]$, where N is the mesh dimension in the CFD context. In POD, the vectors $\mathbf{u}_i$ are called snapshots, and we refer to $\mathbf{S}$ as the snapshot matrix. Usually, each snapshot is associated to a given parameter of the problem, as can be seen in Section 2.1.2.

Successively, a eigenproblem for $\mathbf{S}$ is solved, often through an SVD, or other decomposition techniques. A set $\boldsymbol{\Psi} = \{\boldsymbol{\psi}_j\}_{j=1,\dots,n_{modes}}$ of orthonormal linearly independent generators of $\mathbf{S}$ is found

$$\boldsymbol{\Psi} \xleftarrow{SVD} \mathbf{S} \tag{1}$$

so that for all the elements $\boldsymbol{\psi}_j$ of the basis, called modes, is true that

$$\mathbf{S}\mathbf{S}^T \boldsymbol{\psi}_j = \lambda_j^2 \boldsymbol{\psi}_j. \tag{2}$$

$\lambda_j \in \mathbb{R}$ is the singular value of $\mathbf{S}$ associated to the $\boldsymbol{\psi}_j$ mode, which has the same dimensionality of a snapshot, namely $\boldsymbol{\psi}_j \in \mathbb{R}^N, \forall j = 1, \dots, n_{modes}$. A common practice is to truncate the basis, and the number of modes is reduced to r, often having significantly fewer modes than snapshots, so that $r < n_{modes} \leq n$. For this truncation, different criteria exist, but the most common approach is to sort in decreasing order all the modes from the most energetic one and put a threshold to the POD energy ϵ, defined as

$$\epsilon(r) = \frac{\sum_{j=1}^{r} \lambda_j^2}{\sum_{j=1}^{n} \lambda_j^2}. \tag{3}$$

The reduced number of modes r is set so that $\epsilon(r) \leq$ threshold. Even though truncating the basis $\boldsymbol{\Psi}$ means to waste some potential information, the lowest energy modes are associated to high-frequency behaviors, sometimes related to data noise.

Once the modes are obtained, an encoding of the snapshots can be performed. To do so, the snapshots have to be projected on the reduced basis $\boldsymbol{\Psi}$, obtaining a scalar value α for each snapshot and for each mode. Let us consider the snapshot $\mathbf{u}_i$ and project it on the modes $[\boldsymbol{\psi}_1, \dots, \boldsymbol{\psi}_r]$

$$\alpha_j^i = \mathbf{u}_i \cdot \boldsymbol{\psi}_j \quad \forall j = 1, \dots, r \tag{4}$$

where $\cdot$ represents the scalar product. The set $\{\alpha_j^i\}_{j=1,\dots,r}$ is the POD representation of the snapshot $\mathbf{u}_i$. Since $r \ll N$, where N was the mesh dimension, the POD operate a substantial compression of the information with a controlled trade-off on the accuracy. The approximation of a snapshot with the POD basis becomes

$$\mathbf{u}_i \sim \mathbf{u}_i^* = \sum_{j=1}^{r} \alpha_j^i \boldsymbol{\psi}_j \tag{5}$$

2.1.2. Non-intrusive reduced order model

Historically, POD snapshots are a function of time t, so it is common to associate them to it. Each snapshot $\mathbf{u}(t)$ is representative of a certain instant of time t, and the snapshot matrix is given by $\mathbf{S} = \{\mathbf{u}_{t=0}, \dots, \mathbf{u}_{t=T}\}$. If we generalize this concept, switching from the time to a latent space parametrized by $\boldsymbol{\mu} \in \mathbb{R}^d$, where d is the number of parameters of the problem, the snapshot matrix now becomes $\mathbf{S} = \{\mathbf{u}(\boldsymbol{\mu}_1), \dots, \mathbf{u}(\boldsymbol{\mu}_n)\}$. Therefore, the POD coefficients α_j^i associated to the training snapshots can be written as a function of the parameter $\boldsymbol{\mu}$, since for each mode $j = 1, \dots, r$

$$\alpha_j(\boldsymbol{\mu}) = \mathbf{u}(\boldsymbol{\mu}) \cdot \boldsymbol{\psi}_j \tag{6}$$

However, $\mathbf{u}(\boldsymbol{\mu})$ is known only for the n training snapshots in $\mathbf{S}$. As a result, we have information of the functions $\alpha_j(\boldsymbol{\mu})$ only for specific parameter values $\boldsymbol{\mu}$.

To approximate the POD coefficient function $\alpha_j(\boldsymbol{\mu})$ it is possible to use a regression model $\hat{\alpha}_j(\boldsymbol{\mu})$. The model is fit on the known values $\alpha_j(\boldsymbol{\mu}_i)$ for the parameters $\mu_i \forall i = 1, \dots, n$, and the process can be repeated for each POD coefficient $\alpha_j(\boldsymbol{\mu}) \forall j = 1, \dots, r$ [58].

With the approximated POD coefficients $\hat{\alpha}_j(\boldsymbol{\mu})$, unknown solutions $\mathbf{u}(\boldsymbol{\mu}) \notin \mathbf{S}$ can be reconstructed, similarly to Eq. (5)

$$\mathbf{u}(\boldsymbol{\mu}) \sim \sum_{j=1}^{r} \hat{\alpha}_j(\boldsymbol{\mu}) \boldsymbol{\psi}_j \tag{7}$$

Different techniques can be employed to approximate the POD coefficient functions, such as GPs, NNs or RBFs.

2.2. Multi-fidelity

To achieve the objective of this research, we employ a multi-fidelity regression model [10,11], a framework appealing for its cost-efficiency when studying complex problems due to its ability to balance computational resources and accuracy. This methodology leverages LF information to enhance the HF representation of a given phenomenon. Often, LF data derives from strong approximations of the HF formulation of the problem and, by itself, it is not suitable for the purpose. On the other hand, the associated cost lowers significantly, when it is compared to the HF information.

For the purposes of this research, a kriging-based multi-fidelity regression model has been selected, namely the aforementioned NARGP model, presented in [15].

2.2.1. Gaussian process regression

GP regression, also known as kriging, is an approximation model of a function $y(\mathbf{x})$, with $\mathbf{x} \in \mathbb{R}^d$, given a set of observations of y for different values of the latent variable $\mathbf{x}$. The main idea is to fit a GP on some known data, where a GP is collection of random variables, any finite number of which have a joint Gaussian distribution [59]. A GP is fully defined by a covariance function $k(\mathbf{x}, \mathbf{x}')$, also known as kernel, and a mean function $m(\mathbf{x})$. A common notation for GP is

$$y \sim \mathcal{GP}\left(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')\right) \tag{8}$$

A kernel is a positive semi-definite function, representing the covariance of two random variables belonging to the stochastic process. Squared Exponential kernel and Matern 3/2 kernel are two frequent choices.

$$k_{Mat32}(\mathbf{x}, \mathbf{x}') = \sigma^2 \left(1 + \frac{\sqrt{3}\|\mathbf{x} - \mathbf{x}'\|}{l}\right) \exp\left(\frac{\sqrt{3}\|\mathbf{x} - \mathbf{x}'\|}{l}\right) \tag{9}$$

$$k_{SE}(\mathbf{x}, \mathbf{x}') = \sigma^2 \exp\left(-\frac{1}{2}\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{l^2}\right) \tag{10}$$

where σ^2, l are the variance and the lengthscale of the GP, which are hyperparameters of the model. Using automatic relevance determination [59] is possible to have anisotropic kernels with d lengthscale hyperparameters, where d is the dimension of the parameter $\mathbf{x}$, and improve the flexibility of the GP. To do so, we can consider a generic distance function r_M between two locations $\mathbf{x}, \mathbf{x}'$

$$r_M(\mathbf{x}, \mathbf{x}') = \sqrt{(\mathbf{x} - \mathbf{x}')^T M (\mathbf{x} - \mathbf{x}')} \tag{11}$$

where M is a semi-definite positive matrix. Now Eqs. (10) and (9) can be rewritten in a more general form, such as

$$k_{Mat32}(\mathbf{x}, \mathbf{x}') = \sigma^2 \left(1 + \sqrt{3} r_M(\mathbf{x}, \mathbf{x}')\right) \exp\left(\sqrt{3} r_M(\mathbf{x}, \mathbf{x}')\right) \tag{12}$$

$$k_{SE}(\mathbf{x}, \mathbf{x}') = \sigma^2 \exp\left(-\frac{1}{2} r_M(\mathbf{x}, \mathbf{x}')^2\right) \tag{13}$$

For the purpose of this work we consider only diagonal matrices M, where $[M]_{i,i} = 1/l_i^2$ and l_i is the ith lengthscale hyperparameter. If $l_i = l \quad \forall i = 1, \dots, d$ then the anisotropic kernels in Eqs. (12) and (13) are the same of isotropic kernels in Eqs. (9) and (10). On the other hand, the mean function assumes real values, and represents the expected values of the GP random variables. To train the actual GP regression model, a maximum likelihood estimation (MLE) problem for the hyperparameters, based on the training points, is solved.

Usually there is no prior knowledge on the mean function shape, therefore we can assume without loss of generality its prior to be $m(\mathbf{x}) = 0$. As a consequence, the choice of the kernel formulation and its hyperparameters values is pivotal. Usually, lengthscale hyperparameters are the most challenging to set. Similarly, a well posed optimization problem for the MLE is crucial: to find the best hyperparameters configuration it is mandatory to dial the hyperparameters constraints and choose the proper optimization algorithm set-up. Common choices for the algorithm are gradient-based algorithms, since the log-likelihood is easy to differentiate and leads to faster convergence. However, it should be kept in mind that poor hyperparameter constraining can have a heavy impact on gradient descend-like algorithms. A noise hyperparameter can be added alongside the kernel to handle noisy data.

Once the MLE has been performed, the shape of the kernel is fixed and, assuming without loss of generality a zero mean prior, the GP is fully defined.

Prediction. Let us call $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_n] \in \mathbb{R}^{n \times d}$ the matrix of training inputs, and $\mathbf{y} = [y(\mathbf{x}_1), \dots, y(\mathbf{x}_1)] \in \mathbb{R}^n$ the vector of training outputs, while $\mathbf{X}^* \in \mathbb{R}^{n^* \times d}$ and $\mathbf{y} \in \mathbb{R}^{n^*}$ are the input and output matrices for the n^* unknown target values. If we want both $\mathbf{y}$ and $\mathbf{y}^*$ to belong to the same GP, they have a joint Gaussian distribution so that

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{y}^* \end{bmatrix} = \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} k(\mathbf{X}, \mathbf{X}) & k(\mathbf{X}, \mathbf{X}^*) \\ k(\mathbf{X}^*, \mathbf{X}) & k(\mathbf{X}^*, \mathbf{X}^*) \end{bmatrix}\right) \tag{14}$$

where the kernel k applied to a set of inputs, such as $\mathbf{X}, \mathbf{X}^*$, gives back the covariance matrix for the given points. Since the kernel shape is known, and the inputs for both training and target points $\mathbf{X}, \mathbf{X}^*$ are known too, the unknown posterior distribution of $\mathbf{y}^*$, can be computed analytically

$$\begin{aligned} &\mathbf{y}^* | \mathbf{X}^*, \mathbf{X}, \mathbf{y} = \mathcal{N}(k(\mathbf{X}^*, \mathbf{X}) k(\mathbf{X}, \mathbf{X})^{-1} \mathbf{y}, \\ &k(\mathbf{X}^*, \mathbf{X}^*) - k(\mathbf{X}, \mathbf{X}^*) k(\mathbf{X}, \mathbf{X})^{-1} k(\mathbf{X}^*, \mathbf{X})) \end{aligned} \tag{15}$$

where with | we refer to the conditional probability. The mean of the posterior distribution in Eq. (15) represents the approximation of the target outputs $\mathbf{y}^*$, while the principal diagonal of the covariance matrix $k(\mathbf{X}^*, \mathbf{X}^*)$ gives the variance associated to the prediction for each target.

2.2.2. NARGP

In this work we used the NARGP [15] model for the multi-fidelity approximation of the POD coefficients. This model is based upon GP regression models, detailed in Section 2.2.1. It can be seen as an evolution of cokriging models [10,11], able to handle non-linear relationships between LF and HF information.

The NARGP has a recursive structure, so an arbitrary number of fidelity levels can be considered. For the sake of simplicity, we will consider only two fidelity levels, LF and HF, respectively. The LF and HF models take the same inputs $\mathbf{x}$, and we call their outputs $y_{LF}(\mathbf{x}), y_{HF}(\mathbf{x})$ with $\mathbf{x} \in \mathbb{R}^d$. We will assume that the LF information is more abundant than the HF data, and the LF and HF training points are nested, even though the latter is not mandatory.

$$\mathbf{X}_{LF} = \{\mathbf{x}_i\}_{i=1,\dots,n_{LF}}$$
$$\mathbf{X}_{HF} = \{\mathbf{x}_j\}_{j=1,\dots,n_{HF}} \subset \mathbf{X}_{LF}$$
$$n_{LF} \gg n_{HF}$$

First of all, to train a NARGP, a single fidelity GP regression needs to be performed on the LF information. With the LF model $\hat{y}_{LF}(\mathbf{x}) \sim y_{LF}(\mathbf{x})$ it is possible to approximate the LF function in every point of the latent variables domain. Consequently, we can evaluate the posterior mean of the LF function in the HF training points, obtaining

$$\mathbf{m}_{LF}^{HF} = \hat{y}_{LF}(\mathbf{X}_{HF}) \tag{16}$$

where here $\mathbf{m}_{LF}$ stands for the LF GP posterior mean. For the HF model, $\mathbf{m}_{LF}^{HF}$ is exploited to enrich its input. This input will be:

$$\mathbf{XX}_{HF} = [\mathbf{X}_{HF}, \mathbf{m}_{LF}^{HF}] \in \mathbb{R}^{n_{HF} \times d+1} \tag{17}$$

In order to handle this multi-fidelity input $\mathbf{XX}_{HF}$, a special kernel for the HF model is required. Since products and sums of positive semi-definite functions are in turn positive semi-definite, the combination of kernel

$$k_{HF}((\mathbf{x}, \mathbf{m}_{LF}(\mathbf{x})), (\mathbf{x}', \mathbf{m}_{LF}(\mathbf{x}'))) = k_\rho(\mathbf{x}, \mathbf{x}') \cdot k_f(\mathbf{m}_{LF}(\mathbf{x}), \mathbf{m}_{LF}(\mathbf{x}')) + k_\delta(\mathbf{x}, \mathbf{x}') \tag{18}$$

is still a valid kernel. In general, k_ρ and k_δ will be modeled with anisotropic kernels, while k_f has to be isotropic since it handles 1D inputs. It is possible to notice that different dimensions of the input are managed by different parts of the kernel. Through the optimization of the kernel shape solving the MLE problem, the HF GP is able to learn the relationship between LF and HF data without making any prior assumption on the relation between the two. On the other hand, the complexity of the model is relatively high due to a more complex kernel and therefore more hyperparameters to optimize.

Prediction. The advantage of optimizing a kernel which exploits also LF outputs, stays in the prediction. When we want to find the prediction for a set of unknown input values $\mathbf{X}^*$, it is possible to evaluate the LF output from the known approximation $\hat{y}_{LF}(\mathbf{X}^*)$. Since the model has already been trained, the process is straightforward, as in Section 2.2.1. The LF prediction now can be added to the actual input $\mathbf{X}^*$, similarly to what has been done in Eq. (17), obtaining the composed input $\mathbf{XX}^* = [\mathbf{X}^*, \hat{y}_{LF}(\mathbf{X}^*)]$. This input is coherent to the kernel k_{HF} defined in Eq. (18), and the prediction can be evaluated. Due to the structure of the kernel, the addition of LF information to the input leads to better prediction in unknown zones of the parameter domain, if compared to single fidelity approaches.

Keeping in mind that there is a certain uncertainty associated to the LF prediction, quantified by the variance of the GP, it is important to sample a certain amount of times from the joint distribution of the LF GP, instead of using the mean of the LF GP as an approximation of the LF posterior mean. By doing so it is possible to properly propagate the uncertainty in the LF model.

It is worth noticing that for NARGP, and for GPs in general, it is not always feasible to have large training data sets. According to Eq. (15), each time the model has to predict a new output it has to load all training data. Therefore, these regression models are better suited for modest databases, while POD-based ROMs are more convenient for distributed quantities.

2.3. POD multi-fidelity ROM

The proposed POD multi-fidelity ROM is a multi-fidelity extension of NI POD-based ROMs. The multi-fidelity concept concern both the POD computation and the training of the regression models used for POD coefficients approximation.

In the flowchart in Fig. 1, the model structure is presented. First of all, a parameterization is needed. Usually, in the field of fluid dynamics, latent variables commonly employed include different boundary conditions, geometrical parameterization, time, or physical properties of the fluid. An adequate Design of Experiment (DoE) is mandatory to ease the model training. To this end, techniques such as Latin Hypercube Sampling (LHS), random sampling, or factorial methods are often applied. These allow a better exploration of the latent variable domain, simultaneously reducing the computational cost associated to the offline ROM's phase.

Then, a subset of the designs will be evaluated with the HF solver, creating the HF snapshots. With the LF solver, the LF snapshots are obtained in a similar way. Usually, design configuration associated to the HF database are evaluated with both the solvers, since multi-fidelity regression models benefit from nested low-high fidelity designs.

Subsequently, the POD can be computed on the obtained snapshots. In light of different solution grids between different fidelity levels or due to different geometrical parameterizations, a mapping on a reference mesh of the solutions might be required. POD-like methods call for coherent snapshots to assemble the snapshot matrix $\mathbf{S}$, meaning that all the results must came from meshes with the same amount of nodes and with a fixed connectivity. Otherwise, the POD cannot be computed, or it may produce non-significant results. The mapping will be detailed in Section 2.3.1. Under the hypothesis that all snapshots are defined on the same reference mesh, it is possible to compute the POD. To accomplish this, there are two possibilities: use only the HF snapshots to compute the POD, or use a mixture of LF and HF snapshots. Inevitably, in the first case the representativeness of the POD basis occurs to be limited. On the other hand, if LF snapshots were to be added to the matrix of snapshots, the subspace to which the HF snapshots belong is better spanned, given a LF solver capable to

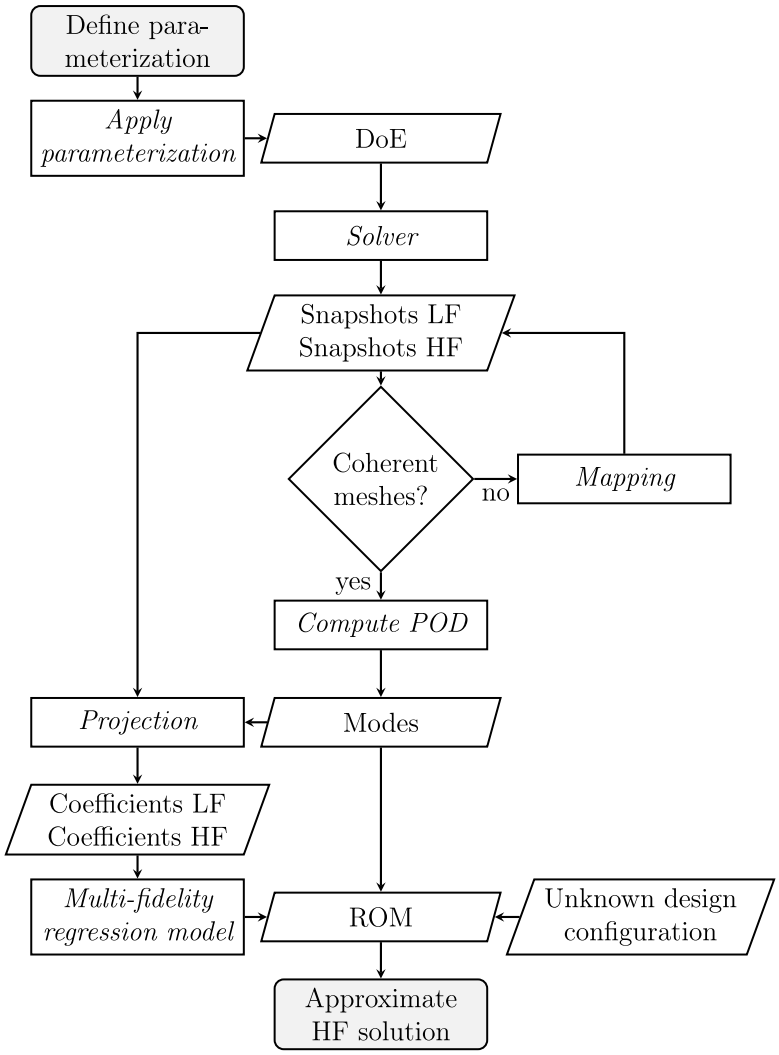


Fig. 1. Flowchart representing the structure of the proposed POD multi-fidelity ROM.

describe at least partially the problem. Nevertheless, since LF snapshots by definition are less informative, part of the HF information might be lost in the process. Keeping in mind that multi-fidelity methods are particularly effective when dealing with sparse HF databases, in most cases, it is beneficial to prefer the mixture of low and high fidelity snapshots. In Sections 3 and 4 this behavior will be deeply discussed.

Once the POD modes are determined, it is possible to project both the LF and HF snapshots onto the basis, obtaining the LF and HF representation of the POD coefficients, respectively. Then a multi-fidelity map from the latent space variables to the HF POD coefficients can be estimated with the NARGP. This process is repeated for each mode in the POD basis. Owing to the multi-fidelity regression, a better approximation of the HF POD coefficients is possible. Improvements are expected especially for the first modes coefficients, since it is more likely to find a relationship between their LF and HF POD coefficients. The underlying reason appears to be that the most energetic modes are associated to more macroscopic physical phenomena, that both LF and HF solver usually are able to catch, even-though with some slight differences. On the other hand, high frequency behaviors, spanned by the high frequency modes, are prerogative of HF snapshots, while LF snapshots have from few to zero related information.

Since for the less energetic modes coefficients a multi-fidelity model might not be required, a single-fidelity equivalent can be used instead. From a practical standpoint, a naive criterion based on the correlation between nested LF-HF designs can be used to decide in advance how many POD modes coefficients might benefit from a multi-fidelity modeling. However, it should be kept in mind that there is no universal approach which is valid for every problem. Since a properly trained NARGP should perform at least as well as a single-fidelity GP in absence of relationship between different fidelity levels, the aforementioned criterion is not mandatory, even though it lowers the online computational cost being NARGP heavier than single-fidelity GPs. If there is no relationship between LF and HF data, looking to Eq. (18), we expect the amplitude of $k_\rho \cdot k_f$ tends to 0 during the optimization of the hyperparameters. The amplitude is given by the product of the variances of the two kernels k_ρ, k_f, and the kernel of the NARGP transition to a kernel of a simple single fidelity GP regression model, such as in Eq. (19), ignoring the LF output.

$$k_{HF} = \cancel{k_\rho \cdot k_f} + \overset{0}{} k_\delta \sim k_\delta \tag{19}$$

After the coefficients models are trained, the prediction of new and unknown design configurations is straightforward. For each mode, a multi or single-fidelity prediction of the POD coefficient is evaluated, and subsequently a linear combination of the modes is performed, where the weights are given by the approximated POD coefficients $\hat{\alpha}_j(\boldsymbol{\mu}) \forall j = 1, \dots, r$. If we assume that the first $j^* < r$ modes are relevant for the multi-fidelity approximation, for any value of $\boldsymbol{\mu}$ we obtain that

$$\mathbf{u}(\boldsymbol{\mu}) \sim \sum_{j=1}^{j^*} \hat{\alpha}_{j,m}(\boldsymbol{\mu}) \boldsymbol{\psi}_j + \sum_{j=j^*+1}^{r} \hat{\alpha}_{j,s}(\boldsymbol{\mu}) \boldsymbol{\psi}_j \tag{20}$$

where $\hat{\alpha}_{j,m}, \hat{\alpha}_{j,s}$ refer to the multi-fidelity and single-fidelity POD coefficient models, respectively. As it will be discussed in Section 3.2, not all the POD modes coefficients can leverage the benefits of a multi-fidelity model since usually the last modes do not carry any significant information for the LF snapshots. Therefore, it is possible to model the last POD modes coefficients with single-fidelity models as in Eq. (20) without affecting significantly the approximation.

NARGP motivation. The choice of using NARGP over other multi-fidelity regression models based on GPs, such as Cokriging [10] or hierarchical Kriging [13], is due to the NARGP flexibility. First of all, the LF-HF POD coefficient correlations will be different depending on the considered problem, and are affected by the LF-solver choice too. Secondly, each POD mode will show, in general, a different correlation in its LF-HF POD coefficients. Therefore, unless there is prior knowledge to similar use-cases, it is not possible to make a priori assumption on the LF-HF POD coefficients correlations. Indeed, this can affect the capabilities of models like Cokriging, which can leverage only close to linear correlations. On the other hand, NARGP is meant to handle non-linear correlations between fidelities, overcoming this limitation [15,60]. As a consequence, NARGP has more hyperparameters than Cokriging-like models to exploit more complex LF-HF relationships, being more prone to overfit. With this in mind, the flexibility offered by NARGP over other multi-fidelity GP-based models is particularly fit to handle real-world problems.

2.3.1. Mapping

The adoption of the mapping strategy is pursued to have all snapshots with a coherent discretization. To perform the POD, the cardinality of each snapshot must be identical. In addition, the indexing must be done so that the connectivity is preserved between different snapshots. Here the mapping approach employed in this work will be discussed, and an overview of the whole process can be found in Algorithm 1.

In particular, there will be a focus on an external aerodynamics problem for a road vehicle that has completely different meshes between both different designs and different fidelity levels. For instance, the deformations of the car are consequent to its geometrical parameterization, while more refined grids account for the different fidelity levels. Due to the geometrical deformations that have been introduced, the proposed mapping approach is bound to analogue external aerodynamic problems.

Given a finite volume method CFD simulation, the values of the flow field are known in the internal cells centers, and on the face centers for the boundary ones. Without loss of generality, we suppose that the quantities of interest for the ROM are distributed on the external surface of the vehicles, such as the pressure field.

The mapping consists of three main steps: the generation of a common reference mesh; the morphing of an undeformed triangulated geometry; the interpolation of the solution field onto the reference nodes. Being the relation between parameterization and deformation explicitly known, this method differs from geometry registration techniques [61].

Since the boundary fields of all car designs need to be coherent, the results have been mapped on a reference mesh. The reference mesh has to be appropriate for both LF and HF snapshots. Moreover, since each vehicle grid has been generated independently, the position of the nodes is unpredictable. Therefore, it is advisable to have a fine reference mesh, even more refined than the HF ones. Otherwise, some loss of information might occur. Regarding the shape of the reference car, the baseline geometry was used.

Let $\mathcal{S}_{car} \subset \mathbb{R}^3$ be the external surface of the baseline car, and $\boldsymbol{\Phi}(x;\boldsymbol{\mu})$ the function that deforms its points, given a geometrical parameterization $\boldsymbol{\mu} \in \mathbb{R}^d$ and the coordinates $x \in \mathcal{S}_{car}$, the deformed geometry is given by:

$$\mathcal{S}'_{car} = \left\{ x' : x' = \boldsymbol{\Phi}(x;\boldsymbol{\mu}), \quad \forall x \in \mathcal{S}_{car} \right\} \tag{21}$$

where $\mathcal{S}'_{car}$ represents a set of points $x' \in \mathbb{R}^3$ that belong to the surface of the deformed car. In Fig. 2, there is a schematic representation of the effect of a generic deformation $\boldsymbol{\Phi}$ when applied on a mesh.

In this work, the deformations were obtained with Optimad's software *mimic*[1] for each different car. This software is able to manipulate surface features and propagate the deformation field smoothly around the features themselves, while preserving imposed geometrical constraints. To do so, a Laplacian equation of the deformations is solved. Consequently, the acquired deformed designs manufacturability persists unaltered. An example of its usage is presented in Fig. 5 for the proposed test case, where six feature edges are displaced, propagating the deformation to the rest of the vehicle surface. For further computations, each car target field (e.g. pressure) is interpolated from the boundary face centers to the nodes of the solution grid.

When the computational grid does not share the same connectivity through each design and geometrical deformations are involved, the knowledge of the deformation field, or a surrogate of its, is mandatory to train a ROM. If the deformations are known, this procedure can be generalized and extended to any similar problem.

For a given deformed car, the reference mesh points are projected on the deformed car, exploiting the knowledge of the deformation field, as in Fig. 3. The resulting set of points will then include both the reference mesh points and the projected ones, living on the deformed geometry.

Each projected point of the reference mesh will now belong to a single face of the deformed car's solution grid. With respect to the vertices of this face, the reference point's barycentric coordinates can be evaluated by:

$$\left[w_P^{V_i} \right]_{i=1,\dots,N_V} = \left[\frac{1}{\| x_{V_i} - x_P \|_2} \right]_{i=1,\dots,N_V} \tag{22}$$

where V_i represent the ith vertex of the face, N_V is the number of vertices of the face, with x we indicate the position of a point in space, and $\left[w_P^{V_i} \right]_i$ are the barycentric coordinates of P in respect of the vertices of the face.

With the barycentric coordinates of every reference point and the values of the target field, it is possible to interpolate the deformed car target field values onto the reference mesh:

$$p^{ref}(x_P;\boldsymbol{\mu}) = \sum_{i=1}^{N_V} w_P^{V_i} \cdot p^{def}(x_{V_i};\boldsymbol{\mu}) \tag{23}$$

where p^{ref}, p^{def} are respectively the target fields on the reference mesh and the deformed one. By doing so, homogeneous snapshots of the target fields can be collected, even if the number of mesh cells differs, or the grid's connectivity changes. In Fig. 3, it is illustrated the mapping of a node from the reference mesh to the deformed geometry, when having different discretizations.

Both the barycentric interpolation and the geometry deformation techniques are well known and have been used for many applications. In particular, the geometry deformation is based on Free-from deformations, and in the context of ROMs they are used to move the computational mesh nodes according to a deformation field applied to the geometry itself [62]. However, to the authors knowledge, using this techniques to map surface fields from non-coherent boundary tessellations is novel. Therefore, this approach allows to use computational meshes that were generated separately for each design configuration. This enables POD-based workflows, which have strict requirements on their elements order and cardinality, to handle larger geometry deformations.

[1] *mimic*: Geometry parameterization and mesh deformation software. https://www.optimad.it/mimic-geometry-parameterisation-and-mesh-morphing.

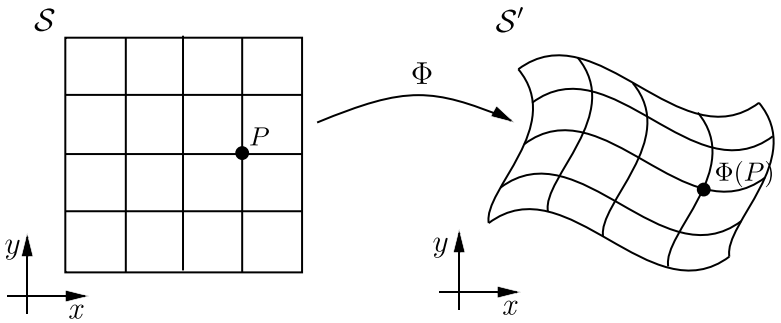


Fig. 2. Schematic representation of the effect of the geometrical deformations on a mesh. Φ represents the function of the transformation, $\mathcal{S}$ the original grid, $P \in \mathcal{S}$ a point, $\mathcal{S}'$ and $\Phi(P)$ are their respective transformations.

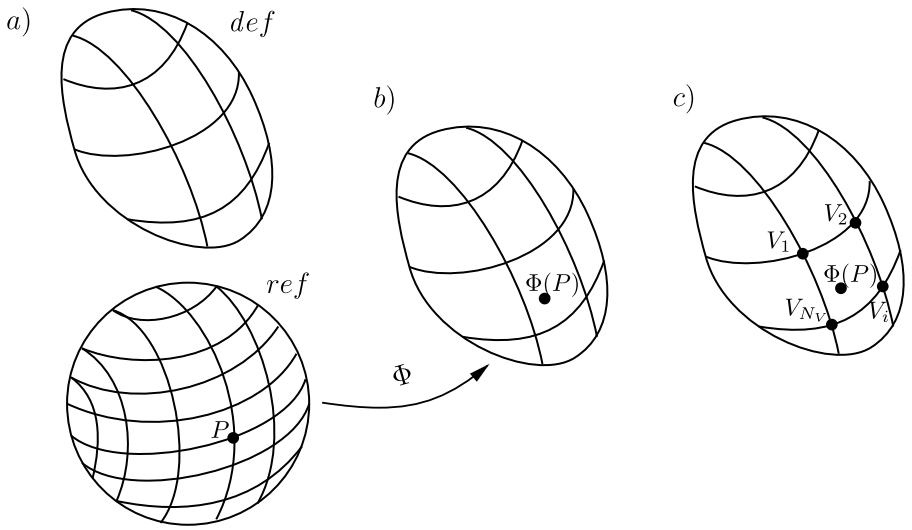


Fig. 3. Schematic representation of the mapping of a reference (*ref*) mesh node to the deformed (*def*) geometry . (*a*) the *ref* and *def* geometries with meshes; (*b*) the transformation of the point P on the *ref* mesh to the *def* geometry; (*c*) vertices of the face to which $\Phi(P)$ belongs to the mesh of the *def* geometry.

Algorithm 1 Overview of the full mapping process

1: Create a database of deformed car geometries (*mimic*)
2: Solve the CFD problem for the deformed cars (*openFOAM*)
3: Define a reference surface mesh from the undeformed vehicle
4: Deform the reference surface mesh
5: Interpolate the results from the CFD solution to the deformed reference mesh
6: Map the solution on the deformed reference mesh back to the undeformed reference mesh

3. Results

3.1. Test case

Parameterization. To validate the model, the DrivAer test case was used. Starting from the fastback configuration of the DrivAer, a total of 6 geometrical deformations were applied. In Fig. 4, there is a representation of the DrivAer in fastback configuration.

As already described in Section 2.3.1, the software *mimic* was used to apply the deformations, which are related to 6 software scalar parameters respectively. These deformations will be representing the geometrical parameterization of the problem, and their effect can be found in Fig. 5. The deformations introduced with *mimic* inherently parameterize the geometry, directly on the triangulated vehicle boundary, not requiring a canonical parametric Computer Aided Design (CAD) software. Thanks to that, the deformation functions Φ for the vehicle meshes are known:

$$\Phi\left(x; [\mu_1, \ldots, \mu_6]\right) = x' \tag{24}$$

where $[\mu_1, \ldots, \mu_6]$ represent the 6 deformations applied to the undeformed DrivAer body.

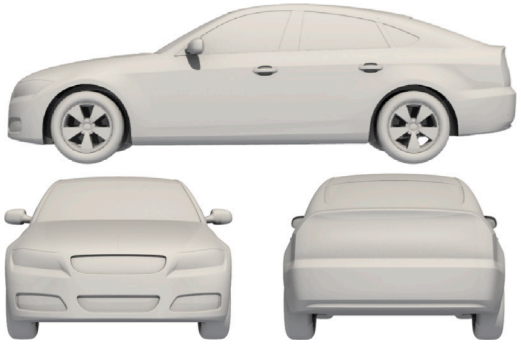


Fig. 4. DrivAer in fastback configuration: on top — side view; on bottom from left to right — front and back views.

Simulation set-up. In this work, we selected as snapshots the pressure fields on the vehicles boundary. To obtain the pressure fields for each deformed car, a CFD simulation has been computed. A box domain around the halved car was considered, as in Fig. 6, with dimensions $10.0L \times 2.2L \times 2.5L$, where L is the car length. The vehicle has been

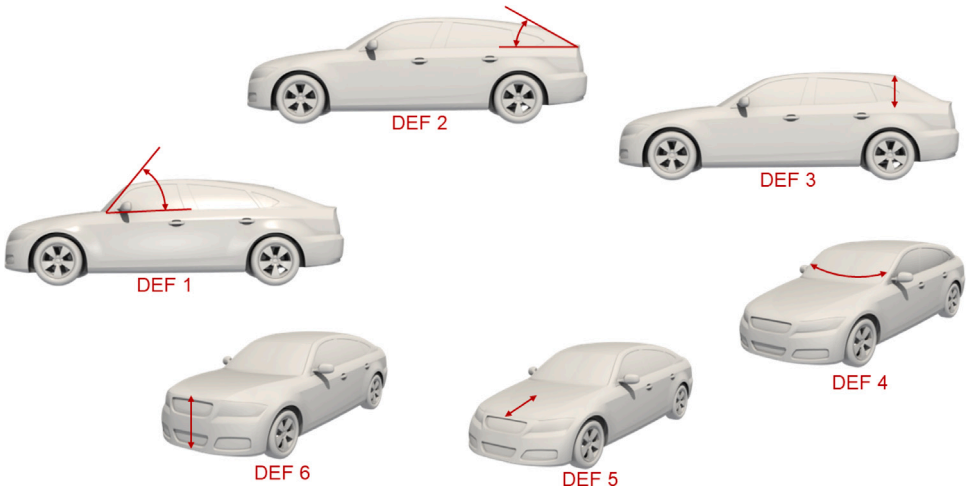


Fig. 5. Representation of car deformations, in clock-wise order from the left: *DEF 1* Front window angle; *DEF 2* Rear window angle, *DEF 3* Roof drop; *DEF 4* Greenhouse angle; *DEF 5* Bumper nose extrusion; *DEF 6* Bumper nose drop.

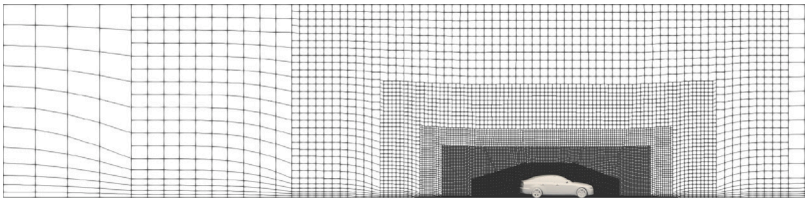


Fig. 6. LF mesh slice, side view.

halved to exploit the geometrical symmetry by its length. The solver set-ups for the LF and HF simulations are identical, except for the mesh discretization: LF meshes have around 1.8 million cells, while the HF ones are up to 6 million.

The incompressible steady-state Reynolds-Averaged Navier Stokes equations are solved together with the Spalart–Allmaras turbulence model [62–64]. This turbulence model is routinely employed in similar industrial design thanks to its simplicity and ease of use. However, some of the considered vehicle designs reflect the behavior typical of a bluff body, therefore, the Spalart–Allmaras model is not always reliable enough to adequately capture the flow sensitivity due to some geometrical changes. In the context of preliminary design choices, exploring many different geometrical configurations is usually more important than a high accuracy of the CFD simulations, which is inevitably expensive to achieve. We decided to accept this accuracy-cost compromise, valuing the capability to discern the general performances of multiple designs.

`simpleFoam`, embedded in the open-source CFD software `openFOAM`, has been employed to solve the set of governing equations. The far field air velocity is set to $U_0 = 38.89$ m/s leading to a Reynolds number based on the length of the model of $\sim 12 \cdot 10^6$. The solution is averaged over the last 500 iterations due to the inherent non-steadiness of the problem, and the pressure fields on the car surface are extracted, as in Fig. 7.

For what concerns the differences between the low- and high-fidelity pressure solutions, Fig. 8 shows an example of the LF solution deviation from the HF one. Given the two snapshots, both representative of the same design, it is possible to see that they are sensible differences due to the different discretization. The different capabilities of the LF and HF solvers can be seen in Fig. 14 too. Here, the LF solver, over-estimates by a good margin the Coefficient of Drag (C_D), defined as

$$C_D = \frac{2D}{A_f \rho U_0^2} \tag{25}$$

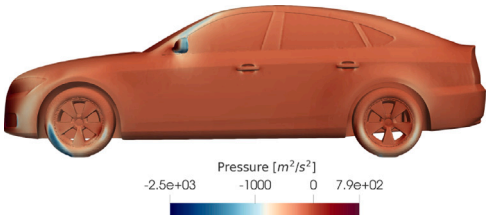


Fig. 7. HF pressure field on the halved car; side view. Pressure is divided by the density.

where D is the pressure contribution to the drag force on the car body and A_f is the frontal area and ρ is the fluid density.

Indeed, Figs. 8 and 14 together show that the LF under-defined computational mesh effect on the LF solution, specifically the inability to capture high frequency behaviors at the HF grid scale, which have a sensitive impact on the car's body pressure distribution.

3.2. Model set-up

In order to compute the POD, several HF snapshots sets were used, starting from a maximum of 120 to a minimum of 20 HF snapshots, using a LHS strategy for the DoE creation. Alongside, 160 LF snapshots were added for the multi-fidelity implementation, again with an LHS strategy. The DoE creation is further discussed in Appendix. Part of the LF snapshots are always matched to the corresponding HF design used, in order to improve the performances of the NARGP model.

A fixed value for the POD energy was considered, with $\varepsilon = 0.999$ according to Eq. (3). This choice allows a fair comparison between the single-fidelity POD ROMs and the proposed multi-fidelity one, as in Section 3.3.

Since the multi-fidelity approach is sensible to the relationship between the LF and HF information, only the coefficients associated

Table 1
Models set up.

Option	NARGP		
	LF model	HF model	Single-fidelity GP
Optimization algorithm	BFGS[a]	BFGS	BFGS
Max iterations	2500	2500	2500
Optimization restarts[b]	10	8	10
Kernel	Matern 3/2	k_ρ : Matern 3/2 k_f : Matern 3/2 k_δ : Matern 3/2	Matern 3/2
Lenghtscale bounds[c]	$0.5 \leq l_i/d_i \leq 30$	k_ρ : $0.5 \leq l_i/d_i \leq 10$ k_f : free k_δ : $0.5 \leq l_i/d_i \leq 10$	$0.5 \leq l_i/d_i \leq 10$
Gaussian noise variance	free	free	free

[a] Broyden–Fletcher–Goldfarb–Shanno algorithm.
[b] Random restarts of the optimization to increase robustness.
[c] The ith lengthscale is called l_i and the ith input variable range of variation's amplitude called d_i; different lengthscales are applied only to anisotropic kernels.

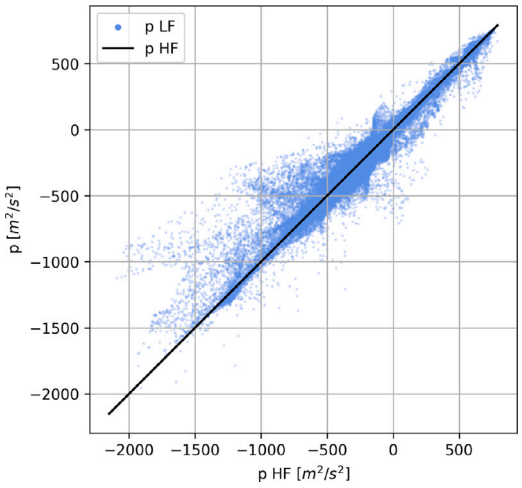


Fig. 8. Scatter plot between HF and LF pressure solutions associated to the same design; each point correspond to the same location on the vehicle's surface. Pressure is divided by the density.

to the first modes are regressed with the NARGP model. This stems from two factors: firstly, the LF snapshots struggle to capture high frequency features, due to the coarser computational grid in respect of the HF case; secondly, the last and less energetic POD modes are often associated to these high frequency behaviors. Consequently, the projection of LF snapshots on the last part of the POD basis will not necessarily be helpful for the multi-fidelity approach. In this work, the first 4 modes coefficients have been dealt with the multi-fidelity model. On the other hand, each of the less energetic modes coefficients have been approximated with single fidelity GP regression models. As stated in Section 2.3, there is no universal criterion to decide how many POD coefficients should be modeled with a multi-fidelity approach, and we decided to use this simple, but effective method. We observed diminishing returns exceeding the first 4 modes in terms of reconstruction error (which is defined in Section 3.3) at the cost of a greater online computational effort.

For each set of snapshots the set up values are similar, except for the lenghtscale hyperparameters, where the lower and upper bounds constraints change with the number of snapshots. A summary of these parameters can be found in Table 1. The single-fidelity GP column refers to the options for the single fidelity model fitted to the POD coefficients relative to less energetic modes. Nonetheless, the same options have been employed also for the single fidelity GP regression model used for comparison.

3.3. Model results

The results of the multi-fidelity POD ROM application to the test case described in Section 3.1 are considered for a total amount of 20, 40, 60, 80, 100, 120 HF snapshots, and a constant number of 160 LF snapshots. The HF snapshots are successive subsets of the starting 120 HF snapshots. A total of 70 HF validation snapshots are reserved for testing purposes. The sampling of the validation snapshots has been performed with a random uniform distribution in the parameter space.

The first metric to consider is the POD projection error, defined as:

$$Err_{prj} = \frac{1}{n_{val}} \sum_{i=1}^{n_{val}} \frac{\left\| \left(\sum_{j=1}^{n_{modes}} (\mathbf{u}_i^{val} \cdot \boldsymbol{\psi}_j) \boldsymbol{\psi}_j \right) - \mathbf{u}_i^{val} \right\|_2}{\left\| \mathbf{u}_i^{val} \right\|_2} \tag{26}$$

where $\mathbf{u}$, $\boldsymbol{\psi}$, n_{val} and n_{modes} represent respectively a generic snapshot, a POD mode, the number of validation snapshots and the number of POD modes, and $\|\cdot\|_2 : \mathbb{R}^N \to \mathbb{R}$ is the L_2 norm in $\mathbb{R}^N$. In the context of this work, the generic snapshot $\mathbf{u}$ corresponds to the pressure distribution on the car body.

This error quantifies the capability of the POD reduced basis to represent the actual validation snapshots. This error metric is independent from the POD coefficients regression models, evaluating the performances of the sole decomposition operator. The projection error trend for different amount of HF training snapshots is represented in Fig. 9.

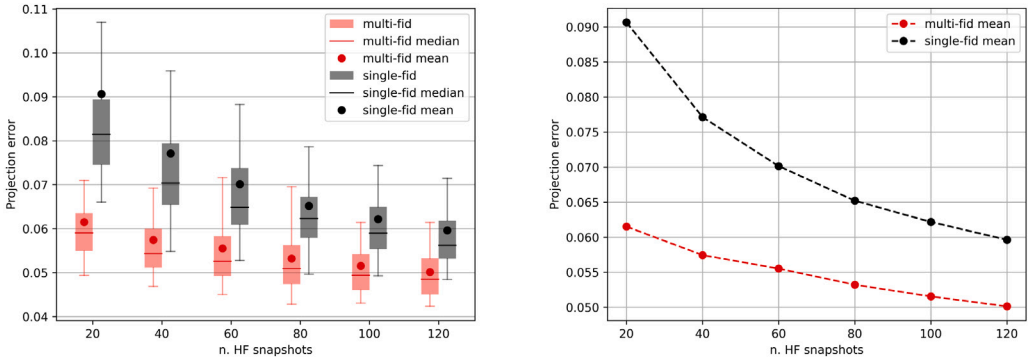
Subsequently, the influence of the POD coefficients regression models can be introduced by considering the reconstruction error

$$Err_{recon} = \frac{1}{n_{val}} \sum_{i=1}^{n_{val}} \frac{\left\| \left(\sum_{j=1}^{n_{modes}} \hat{\alpha}_j(\boldsymbol{\mu}_i^{val}) \boldsymbol{\psi}_j \right) - \mathbf{u}_i^{val} \right\|_2}{\left\| \mathbf{u}_i^{val} \right\|_2} \tag{27}$$

where $\hat{\alpha}_j(\boldsymbol{\mu})$ is the approximation model of the jth POD mode coefficient and $\boldsymbol{\mu}$ is the vector of the independent variables. The reconstruction error trend for different amount of HF training snapshots is represented in Fig. 10. Here, it is possible to evaluate the capabilities of the entire model to approximate an unknown snapshot. The L_2 norm is widely used to evaluate the ROM's error metrics [41,53] and allows to measure the intensity of the absolute error field relative to the original solution.

In Figs. 11 and 12, the detailed distributions of projection and reconstruction errors have been represented for both the extreme configurations with 20 and 120 HF training snapshots. Conversely to Figs. 9(a) and 10(a), outliers are represented too.

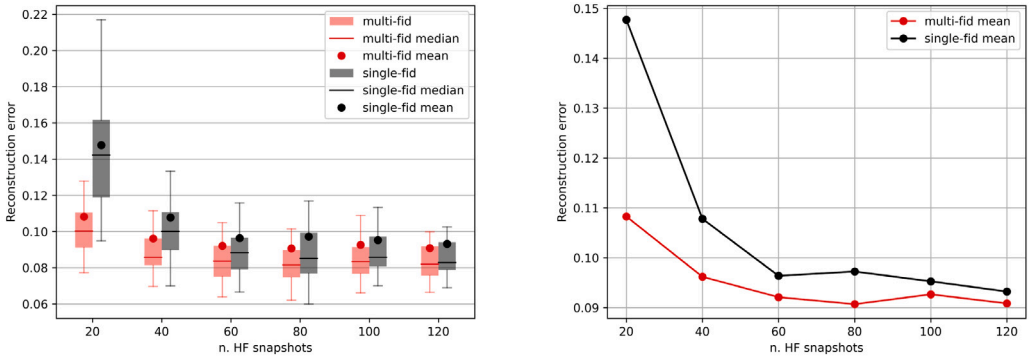
The reconstruction errors of the ROMs in Fig. 12 allow to compare the results of the multi- and single-fidelity ROMs. However, to assess whether or not they are a good approximation of the HF CFD solution, namely the full-order model, the results can be compared with the LF CFD solutions, presented in Fig. 13. Here, the LF solutions are used to directly approximate the HF snapshots reserved for validation, and it



(a) Error boxplots on 70 HF validation snapshots, outliers not represented.

(b) Mean projection error on 70 HF validation snapshots.

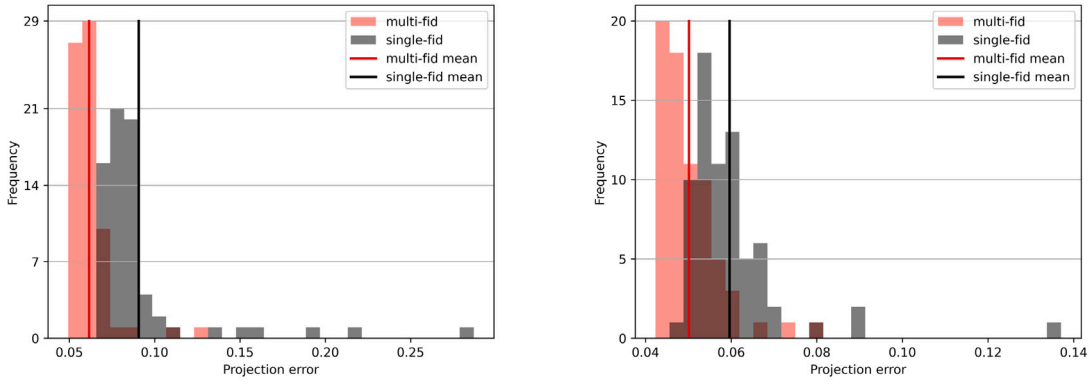
Fig. 9. Projection error comparison between single/multi-fidelity PODs for different numbers of HF snapshots.



(a) Error boxplots on 70 HF validation snapshots, outliers not represented.

(b) Mean reconstruction error on 70 HF validation snapshots.

Fig. 10. Reconstruction error comparison between single-fidelity ROM and multi-fidelity ROM for different numbers of HF snapshots.



(a) 20 HF training snapshots

(b) 120 HF training snapshots

Fig. 11. Comparison of single/multi-fidelity projection error distributions on 70 HF validation snapshots.

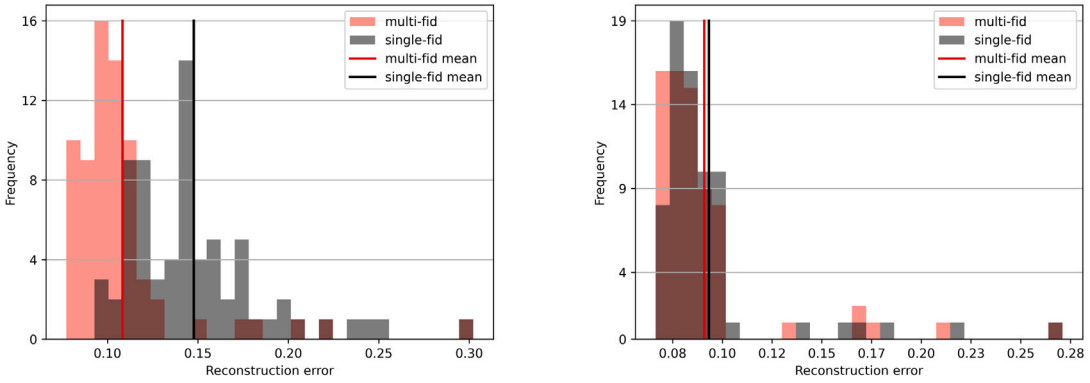
can be seen that both single- and multi-fidelity ROMs are better HF surrogates than the LF solver. To measure the LF approximation error in Fig. 13, the error was computed as in Eq. (28), similarly to Eqs. (26) and (27).

$$Err^{LF} = \frac{1}{n_{val}} \sum_{i=1}^{n_{val}} \frac{\left\| \mathbf{u}_i^{LF,val} - \mathbf{u}_i^{val} \right\|_2}{\left\| \mathbf{u}_i^{val} \right\|_2} \tag{28}$$

where $\mathbf{u}_i^{LF,val}$ is the LF representation of the ith validation snapshot.

With the LF approximation error in Fig. 13, integral quantities related to the pressure deviates as in Fig. 14, where the C_D values are presented for the same designs. Similarly to Fig. 8, Fig. 14 underlines the LF solver difficulty to match the HF solver performances.

Tables 2 and 3 report the time needed for both the CFD simulation and the ROMs. Each CFD solution is 4 times more expensive than the LF one. On the other hand, the ROMs training is at least 1 or 2 order of magnitude less time consuming than a single CFD solution, without taking into consideration the necessary off-line CFD evaluations. Being in the order of milliseconds, the prediction time is negligible for



(a) 20 HF training snapshots (b) 120 HF training snapshots

Fig. 12. Comparison of single/multi-fidelity reconstruction error distributions on 70 HF validation snapshots.

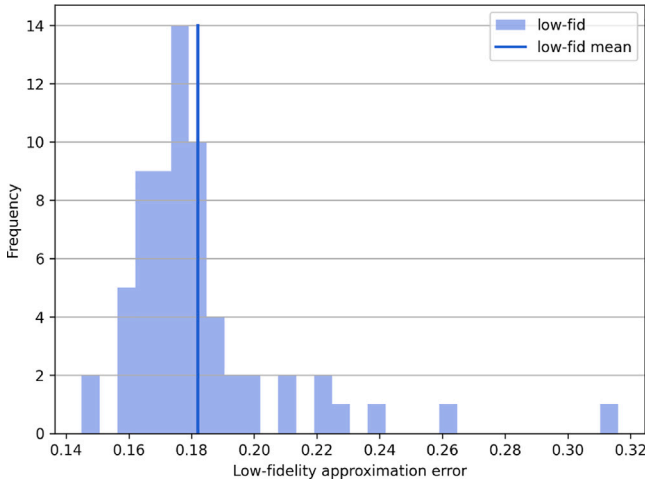


Fig. 13. LF approximation of 70 HF validation snapshots; mean error of 18.2%.

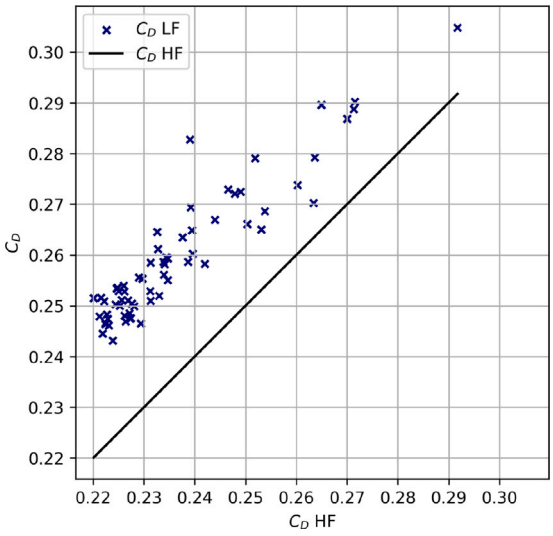


Fig. 14. HF and LF C_D values for the validation designs. The frontal area is kept constant, equal to reference vehicle frontal area.

Table 2
CFD solvers solution time. All calculation on 96 CPU cores (AMD EPYC 7413).

	Low-fidelity		High-fidelity	
	Wall time	Core hours	Wall time	Core hours
Average solution time	24 m 59 s	39.97 h	1 h 39 m 09 s	158.64 h
Time ratio respect LF	1		3.97	

Table 3
ROMs wall time — 20 HF snapshots (plus 160 LF snapshots for the multi-fidelity).

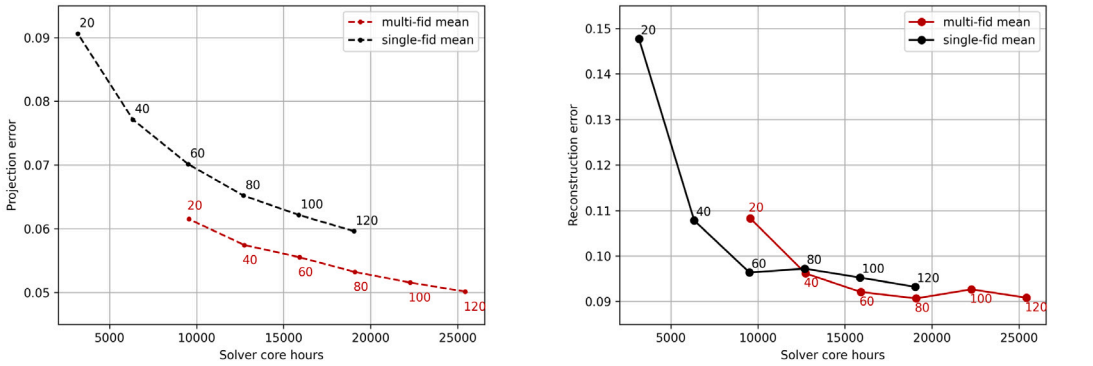
	Single-fidelity	Multi-fidelity
Decomposition time	1.2 s	42.1 s
Training time	24.2 s	2 m 45 s
Number of modes	15	99
Prediction time (1 snapshot)	$5.3 \cdot 10^{-4}$ s	$2.4 \cdot 10^{-3}$ s
Total time	25.4 s	3 m 27 s

both the single, and the multi-fidelity ROM, enabling almost real-time capabilities.

In Table 2 it is possible to see that the major contributor to the ROM time cost is the off-line CFD phase. If we look at the projection and reconstruction errors in respect to the core hours needed for the off-line phase, the results are like in Fig. 15. In terms of projection error, Fig. 15a shows that the mixing of LF and HF snapshots in the POD is beneficial also from an efficiency standpoint. On the other hand, the reconstruction error in Fig. 15b shows that the multi-fidelity approach improves the efficiency of the ROM only after 40 HF snapshots. This is due to the cost of computing 160 LF solutions and it is specific to this test case. Other LF formulation for different test cases can lead to different efficiencies, however, we want to focus on the fact that poor quality LF information can improve the HF representation of the problem. This can be seen in Fig. 15b, and it is even more evident in Fig. 10(b).

4. Conclusion

The purpose of the multi-fidelity POD ROM presented in this work is to improve accuracy of single-fidelity ROMs by means of multi-fidelity strategies, while reducing the amount of HF information needed to train the model. Moreover, the proposed industrial problem allowed to address close to real-world technical challenges, providing a qualitative demonstration of the capabilities of the multi-fidelity POD ROM. Due to the nature of the test-case, all of the previous objectives were pursued while handling non coherent information sources and a wide range of geometrical deformations. In light of the results showed in Section 3.3, several conclusions can be inferred.



(a) Mean projection error vs CFD core hours. (b) Mean reconstruction error vs CFD core hours.

Fig. 15. Projection (a) and reconstruction (b) errors versus the off-line core hours needed for the CFD solutions. Numbers indicate how many HF snapshots are used. All multi-fidelity errors are obtained with the same 160 LF snapshots.

Table A.4
Input parameters range of variation and baseline configuration. Actual values depend on the software definition.

	Range	Baseline	Normalized range
DEF 1	[−0.25, 0.35]	0	[0, 1]
DEF 2	[−0.2, 0.4]	0	[0, 1]
DEF 3	[−0.1, 0.1]	0	[0, 1]
DEF 4	[0.7, 1.2]	1	[0, 1]
DEF 5	[−0.15, 0.15]	0	[0, 1]
DEF 6	[−0.15, 0.15]	0	[0, 1]

First of all, the addition of LF snapshots at the decomposition stage had a great impact on the reduced basis representability of unknown snapshots. This is easily noticeable in Fig. 9, where the single-fidelity POD produces projections between 1% to 3% less accurate than the multi-fidelity POD, where LF snapshots were mixed with the HF ones.

Secondly, not only the overall representation capabilities of the basis improved, but the multi-fidelity POD is able to reduce the projection error also in the outlier configurations. Being the configurations with the highest errors, is advantageous to enhance the modes capability to represent them.

Finally, there is an improvement in the reconstruction error too, especially with fewer HF snapshots which was the actual purpose of using the multi-fidelity POD ROM. The increased accuracy can be seen in Fig. 10, where the multi-fidelity approach reaches lower reconstruction errors with fewer HF snapshots. It is important to observe that with respect to the formulation of the NARGP in Section 2.2.2, the LF snapshots have to be nested with the HF training ones. As mentioned in Section 3.2, 160 LF snapshots were added each time to the n_{HF} HF training snapshots, therefore only $160 - n_{HF}$ LF snapshots actively contributed to better explore the design space. This justifies the superior improvements with fewer HF snapshots and, consequently, the lessened ones when approaching 120 HF training snapshots. Furthermore, this behavior suggests a certain degree of robustness of the POD to the LF information, since it does not deteriorate the quality of the HF representation.

Given all these considerations, the proposed multi-fidelity POD ROM was able to augment the capabilities of analogue single-fidelity NI-ROMs, both in terms of representability and accuracy. The advantages in terms of computational costs are inevitably related to the choice of the LF model, and, as a downside, the snapshots mapping strategy has to be tailored to the specific test-case. In the proposed application the computational cost-accuracy ratio breaks even around 40 HF snapshots, while adding more HF information diminish the returns of the multi-fidelity model with the given LF snapshots. Nevertheless, with this experiment we proved that the HF approximation of a ROM could benefit from otherwise useless LF information.

CRediT authorship contribution statement

Fausto Dicech: Writing – original draft, Visualization, Software, Methodology, Data curation, Conceptualization. **Konstantinos Gkaragkounis:** Writing – review & editing, Software. **Lucia Parussini:** Writing – original draft, Supervision, Conceptualization. **Anna Spagnolo:** Writing – review & editing, Conceptualization. **Haysam Telib:** Writing – review & editing, Resources, Conceptualization.

Funding

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Declaration of competing interest

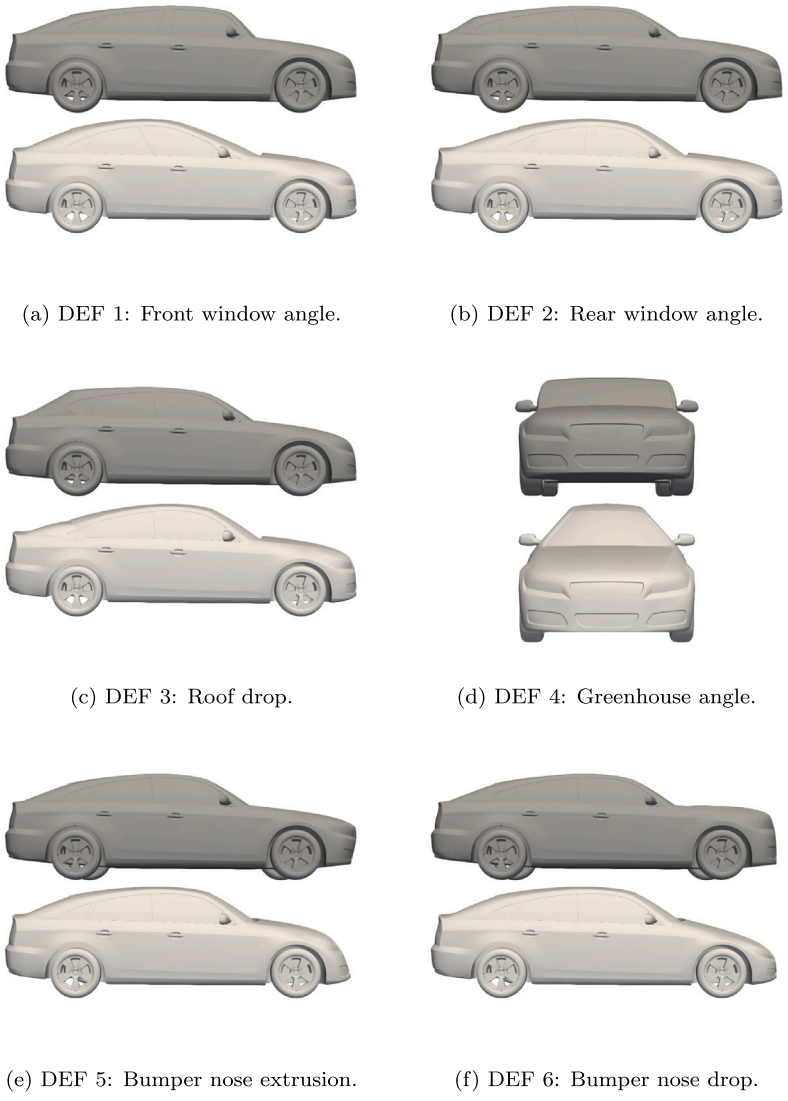
The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix. Design of experiment

The training DoE used for the problem presented in Section 3 is based upon a LHS, where the input parameter values, corresponding to the 6 software deformation values, are normalized in $[0, 1]$, see Table A.4. The same normalization has been used for the validation random DoE.

For each deformation parameter the two extreme configurations are represented in Fig. A.16. Each image was obtained starting from the baseline drivAer model. In Fig. A.16 the darker colored car is obtained when the normalized ith deformation parameter is 1, while the lighter colored one when the normalized ith deformation parameter is equal to 0.

Training DoE. The training DoE was obtained a priori with a LHS strategy, and each design configuration was evaluated with both HF and LF solvers. To add incrementally new designs in a meaningful way when exploring the performances of the ROMs varying the number of HF snapshots, the designs were opportunely ordered. Starting from a first random design, the next designs are added recursively so that for any design, the last one is at maximum distance from the previous ones. This is achieved evaluating the minimal Euclidean distance in $\mathbb{R}^6$ from all the previous designs. Referring to Figures like 9 and 10, the first 20 HF snapshots are contained in the first 40 HF snapshots and so on. Since the LF solutions were obtained for the same design configurations, the 160 LF snapshots used have the first n_{HF} snapshots nested to the n_{HF} HF snapshot used for each ROM.



(a) DEF 1: Front window angle. (b) DEF 2: Rear window angle.

(c) DEF 3: Roof drop. (d) DEF 4: Greenhouse angle.

(e) DEF 5: Bumper nose extrusion. (f) DEF 6: Bumper nose drop.

Fig. A.16. Extreme deformations for all 6 deformation parameters applied separately to the baseline geometry. (a) Front window angle (b) Rear window angle (c) Roof drop (d) Greenhouse angle (e) Bumper nose extrusion (f) Bumper nose drop.

Validaiton DoE. The validation DoE is composed of 70 HF snapshots randomly sampled from the design space, with a uniform random distribution.

The LHS choice for the training DoE represents a good balance between a even coverage of the input parameter domain, while not needing too many desings [65]. This is possible thanks to the pseudo-random nature of the LHS strategy which is adequate also for high dimensional input parameter domains. For the use-case presented in Section 3, the domain is 6-dimensional, so other techniques such as full- and reduced-factorials would not be feasible. On the other hand, a random sampling of the domain, in general, does not fill the space evenly, which is preferable for a training DoE.

Data availability

Data will be made available on request.

References

[1] K. Carlberg, C. Bou-Mosleh, C. Farhat, Efficient non-linear model reduction via a least-squares Petrov–Galerkin projection and compressive tensor approximations, Int. J. Numer. Methods Eng. 86 (2011) 155–181, http://dx.doi.org/10.1002/nme.3050.

[2] A. Scardigli, R. Arpa, A. Chiarini, H. Telib, Enabling of large scale aerodynamic shape optimization through POD-based reduced-order modeling and free form deformation, Comput. Methods Appl. Sci. 48 (2019) 49–63, http://dx.doi.org/10.1007/978-3-319-89988-6_4.

[3] R. Leroux, L. Cordier, Dynamic mode decomposition for non-uniformly sampled data, Exp. Fluids 57 (2016) http://dx.doi.org/10.1007/s00348-016-2165-1.

[4] T. Nonomura, K. Nankai, Y. Iwasaki, A. Komuro, K. Asai, Quantitative evaluation of predictability of linear reduced-order model based on particle-image-velocimetry data of separated flow field around airfoil, Exp. Fluids 62 (2021) http://dx.doi.org/10.1007/s00348-021-03205-8.

[5] Y. Zhang, N. Chen, C. Bronkhorst, H. Cho, R. Argus, Data-driven statistical reduced-order modeling and quantification of polycrystal mechanics leading to porosity-based ductile damage, J. Mech. Phys. Solids 179 (2023) 105386, http://dx.doi.org/10.1016/j.jmps.2023.105386.

[6] M. Castellani, Y. Lemmens, J. Cooper, Parametric reduced-order model approach for simulation and optimization of aeroelastic systems with structural nonlinearities, Proc. Inst. Mech. Eng. G: J. Aerosp. Eng. 230 (2016) 1359–1370, http://dx.doi.org/10.1177/0954410015608888.

[7] C. Perron, D. Sarojini, D. Rajaram, J. Corman, D. Mavris, Manifold alignment-based multi-fidelity reduced-order modeling applied to structural analysis, Struct. Multidiscip. Optim. 65 (2022) http://dx.doi.org/10.1007/s00158-022-03274-1.

[8] D. Xiao, P. Yang, F. Fang, J. Xiang, C. Pain, I. Navon, Non-intrusive reduced order modelling of fluid–structure interactions, Comput. Methods Appl. Mech. Eng. 303 (2016) 35–54, http://dx.doi.org/10.1016/j.cma.2015.12.029.

[9] C. Rowley, T. Colonius, R. Murray, Model reduction for compressible flows using POD and Galerkin projection, Phys. D 189 (2004) 115–129, http://dx.doi.org/10.1016/j.physd.2003.03.001.

[10] M. Kennedy, A. O'Hagan, Predicting the output from a complex computer code when fast approximations are available, Biometrika 87 (2000) 1–13, http://dx.doi.org/10.1093/biomet/87.1.1.

[11] L. Le Gratiet, J. Garnier, Recursive co-kriging model for design of computer experiments with multiple levels of fidelity, Int. J. Uncertain. Quantif. 4 (2014) 365–386, http://dx.doi.org/10.1615/int.j.uncertaintyquantification.2014006914.

[12] Y. Wu, J. Hu, Q. Zhou, S. Wang, P. Jin, An active learning multi-fidelity metamodeling method based on the bootstrap estimator, Aerosp. Sci. Technol. 106 (2020) 106116, http://dx.doi.org/10.1016/j.ast.2020.106116.

[13] Z. Han, R. Zimmermann, Görtz, Hierarchical kriging model for variable-fidelity surrogate modeling, AIAA J. 50 (2012) 1885–1896, http://dx.doi.org/10.2514/1.J051354.

[14] A. Kaps, C. Czech, F. Duddeck, A hierarchical kriging approach for multi-fidelity optimization of automotive crashworthiness problems, Struct. Multidiscip. Optim. 65 (2022) http://dx.doi.org/10.1007/s00158-022-03211-2.

[15] P. Perdikaris, M. Raissi, A. Damianou, N. Lawrence, G. Karniadakis, Nonlinear information fusion algorithms for data-efficient multi-fidelity modelling, Proc. R. Soc. A 473 (2017) http://dx.doi.org/10.1098/rspa.2016.0751.

[16] S. Gano, J. Renaud, J. Martin, T. Simpson, Update strategies for kriging models used in variable fidelity optimization, Struct. Multidiscip. Optim. 32 (2006) 287–298, http://dx.doi.org/10.1007/s00158-006-0025-y.

[17] L. Ng, M. Eldred, Multifidelity uncertainty quantification using non-intrusive polynomial chaos and stochastic collocation, in: 53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference, 2012, http://dx.doi.org/10.2514/6.2012-1852.

[18] Z. Han, S. Görtz, R. Zimmermann, Improving variable-fidelity surrogate modeling via gradient-enhanced kriging and a generalized hybrid bridge function, Aerosp. Sci. Technol. 25 (2013) 177–189, http://dx.doi.org/10.1016/j.ast.2012.01.006.

[19] A. Serani, R. Pellegrini, J. Wackers, C. Jeanson, P. Queutey, M. Visonneau, M. Diez, Adaptive multi-fidelity sampling for CFD-based optimisation via radial basis function metamodels, Int. J. Comput. Fluid Dyn. 33 (2019) 237–255, http://dx.doi.org/10.1080/10618562.2019.1683164.

[20] R. Aydin, F. Braeu, C. Cyron, General multi-fidelity framework for training artificial neural networks with computational models, Front. Mater. 6 (2019) http://dx.doi.org/10.3389/fmats.2019.00061.

[21] M. Guo, A. Manzoni, M. Amendt, P. Conti, J. Hesthaven, Multi-fidelity regression using artificial neural networks: efficient approximation of parameter-dependent output quantities, Comput. Methods Appl. Mech. Engrg. 389 (2022) http://dx.doi.org/10.1016/j.cma.2021.114378.

[22] M. Raissi, G. Karniadakis, Deep multi-fidelity Gaussian processes, 2016, http://dx.doi.org/10.48550/arXiv.1604.07484, arXiv. abs/1604.07484.

[23] X. Meng, G. Karniadakis, A composite neural network that learns from multi-fidelity data: Application to function approximation and inverse PDE problems, J. Comput. Phys. 401 (2019) http://dx.doi.org/10.1016/j.jcp.2019.109020.

[24] K. Cutajar, M. Pullin, A. Damianou, N. Lawrence, J. González, Deep Gaussian processes for multi-fidelity modeling, 2019, http://dx.doi.org/10.48550/arXiv.1903.07320, arXiv. abs/1903.07320.

[25] A. Howard, M. Perego, G. Karniadakis, P. Stinis, Multifidelity deep operator networks for data-driven and physics-informed problems, 2022, http://dx.doi.org/10.48550/arXiv.2204.09157.

[26] L. Lu, R. Pestourie, S. Johnson, G. Romano, Multifidelity deep neural operators for efficient learning of partial differential equations with application to fast inverse design of nanoscale heat transport, Phys. Rev. Res. 4 (2022) 023210, http://dx.doi.org/10.1103/PhysRevResearch.4.023210.

[27] S. Chakraborty, Transfer learning based multi-fidelity physics informed deep neural network, J. Comput. Phys. 426 (2021) 109942, http://dx.doi.org/10.1016/j.jcp.2020.109942.

[28] K. Tian, Z. Li, J. Zhang, L. Huang, B. Wang, Transfer learning based variable-fidelity surrogate model for shell buckling prediction, Compos. Struct. 273 (2021) 114285, http://dx.doi.org/10.1016/j.compstruct.2021.114285.

[29] B. Malouin, J. Trépanier, M. Gariépy, Interpolation of transonic flows using a proper orthogonal decomposition method, Int. J. Aerosp. Eng. 2013 (2013) 928904, http://dx.doi.org/10.1155/2013/928904.

[30] D. Toal, On the potential of a multi-fidelity G-POD based approach for optimization and uncertainty quantification, 2014, http://dx.doi.org/10.1115/GT2014-25184.

[31] M. Mifsud, A. Vendl, L. Hansen, S. Görtz, Fusing wind-tunnel measurements and CFD data using constrained gappy proper orthogonal decomposition, Aerosp. Sci. Technol. 86 (2019) 312–326, http://dx.doi.org/10.1016/j.ast.2018.12.036.

[32] T. Benamara, P. Breitkopf, I. Lepot, C. Sainvitu, Multi-fidelity extension to non-intrusive proper orthogonal decomposition based surrogates, in: ECCOMAS Congress 2016, VII European Congress on Computational Methods in Applied Sciences and Engineering, 2016, pp. 4129–4145, http://dx.doi.org/10.7712/100016.2098.9174.

[33] T. Benamara, P. Breitkopf, I. Lepot, C. Sainvit, P. Villon, Multi-fidelity POD surrogate-assisted optimization: Concept and aero-design study, Struct. Multidiscip. Optim. 56 (2017) 1387–1412, http://dx.doi.org/10.1007/s00158-017-1730-4.

[34] M. Mifsud, D. MacManus, S. Shaw, A variable-fidelity aerodynamic model using proper orthogonal decomposition, Int. J. Numer. Methods Fluids 82 (2016) 646–663, http://dx.doi.org/10.1002/fld.4234.

[35] C. Perron, D. Rajaram, D. Mavris, Development of a multi-fidelity reduced-order model based on manifold alignment, 2020, http://dx.doi.org/10.2514/6.2020-3124.

[36] K. Decker, N. Iyengar, D. Rajaram, C. Perron, D. Mavris, Manifold alignment-based nonintrusive and nonlinear multifidelity reduced-order modeling, AIAA J. 61 (2023) 454–474, http://dx.doi.org/10.2514/1.J061720.

[37] A. Thenon, V. Gervais, M. Ravalec, Multi-fidelity meta-modeling for reservoir engineering - application to history matching, Comput. Geosci. 20 (2016) 1231–1250, http://dx.doi.org/10.1007/s10596-016-9587-y.

[38] A. Bertram, C. Othmer, R. Zimmermann, Towards real-time vehicle aerodynamic design via multi-fidelity data-driven reduced order modeling, in: 2018 AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, http://dx.doi.org/10.2514/6.2018-0916.

[39] X. Wang, J. Kou, W. Zhang, Multi-fidelity surrogate reduced-order modeling of steady flow estimation, Int. J. Numer. Methods Fluids 92 (2020) http://dx.doi.org/10.1002/fld.4850.

[40] H. Yang, S. Hong, G. Wang, Y. Wang, Multi-fidelity reduced-order model for GPU-enabled microfluidic concentration gradient design, Eng. Comput. (2022) http://dx.doi.org/10.1007/s00366-022-01672-z.

[41] M. Kast, M. Guo, J. Hesthaven, A non-intrusive multifidelity method for the reduced order modeling of nonlinear problems, Comput. Methods Appl. Mech. Eng. 364 (2020) 112947, http://dx.doi.org/10.1016/j.cma.2020.112947.

[42] J. Yu, J. Hesthaven, Flowfield reconstruction method using artificial neural network, AIAA J. 57 (2019) 482–498, http://dx.doi.org/10.2514/1.J057108.

[43] H. Kang, Z. Tian, G. Chen, L. Li, T. Chu, Investigation on the nonintrusive multi-fidelity reduced-order modeling for PWR rod bundles, Nucl. Eng. Technol. 54 (2022) 1825–1834, http://dx.doi.org/10.1016/j.net.2021.10.036.

[44] M. Motamed, A multi-fidelity neural network surrogate sampling method for uncertainty quantification, Int. J. Uncertain. Quantif. 10 (2020) http://dx.doi.org/10.48550/arXiv.1909.01859.

[45] X. Meng, H. Babaee, G. Karniadakis, Multi-fidelity Bayesian neural networks: Algorithms and applications, J. Comput. Phys. 438 (2019) http://dx.doi.org/10.1016/j.jcp.2021.110361.

[46] S. Pawar, O. San, P. Vedula, A. Rasheed, T. Kvamsdal, Multi-fidelity information fusion with concatenated neural networks, Sci. Rep. 12 (2022) http://dx.doi.org/10.1038/s41598-022-09938-8.

[47] L. Partin, G. Geraci, A. Rushdi, M. Eldred, D. Schiavazzi, Multifidelity data fusion in convolutional encoder/decoder networks, J. Comput. Phys. 472 (2023) 111666, http://dx.doi.org/10.1016/j.jcp.2022.111666.

[48] M. Torzoni, A. Manzoni, S. Mariani, A deep neural network, multi-fidelity surrogate model approach for Bayesian model updating in SHM, in: European Workshop on Structural Health Monitoring, 2023, http://dx.doi.org/10.1007/978-3-031-07258-1_108.

[49] D. Song, D. Tartakovsky, Transfer learning on multifidelity data, J. Mach. Learn. Model. Comput. 3 (2022) 31–47, http://dx.doi.org/10.1615/JMachLearnModelComput.2021038925.

[50] J. Kou, C. Ning, W. Zhang, Transfer learning for flow reconstruction based on multifidelity data, AIAA J. 60 (2022) 5821–5842, http://dx.doi.org/10.2514/1.J061647.

[51] S. Fresca, A. Manzoni, POD-DL-ROM: Enhancing deep learning-based reduced order models for nonlinear parametrized PDEs by proper orthogonal decomposition, Comput. Methods Appl. Mech. Eng. 388 (2022) 114181, http://dx.doi.org/10.1016/j.cma.2021.114181.

[52] S. Ahmed, P. Stinis, A multifidelity deep operator network approach to closure for multiscale systems, 2023, http://dx.doi.org/10.48550/arXiv.2303.08893.

[53] A. Tiba, T. Dairay, F. Vuyst, I. Mortazavi, J. Ramirez, Non-intrusive reduced order models for partitioned fluid–structure interactions, 2023, http://dx.doi.org/10.48550/arXiv.2306.07570.

[54] P. Theissen, J. Wojciak, K. Heuler, R. Demuth, T. Indinger, N. Adams, Experimental investigation of unsteady vehicle aerodynamics under time-dependent flow conditions - Part 1, SAE Int. (2011) http://dx.doi.org/10.4271/2011-01-0177, ISSN:0148-7191, 2688-3627.

[55] A. Heft, T. Indinger, N. Adams, Introduction of a new realistic generic car model for aerodynamic investigations, SAE Int. (2012) http://dx.doi.org/10.4271/2012-01-0168, ISSN:0148-7191, 2688-3627.

[56] P. Benner, S. Gugercin, K. Willcox, A survey of projection-based model reduction methods for parametric dynamical systems, SIAM Rev. 57 (2015) 483–531, http://dx.doi.org/10.1137/130932715.

[57] L. Sirovich, Turbulence and the dynamics of coherent structures. Part I, II and III, Quart. Appl. Math. 45 (1987) 561–571, http://dx.doi.org/10.1090/qam/910462.

[58] T. Bui-Thanh, M. Damodarany, Proper orthogonal decomposition extensions for parametric applications in compressible aerodynamics, in: Proceedings of the 21st AIAA Applied Aerodynamics Conference, 2003, p. 4213, http://dx.doi.org/10.2514/6.2003-4213.

[59] C. Rasmussen, C. Williams, Gaussian Processes for Machine Learning, MIT Press, 2006, http://dx.doi.org/10.7551/mitpress/3206.001.0001.

[60] L. Brevault, M. Balesdent, A. Hebbal, Overview of Gaussian process based multi-fidelity techniques with variable relationship between fidelities, application to aerospace systems, Aerosp. Sci. Technol. 107 (2020) http://dx.doi.org/10.1016/j.ast.2020.106339.

[61] T. Taddei, L. Zhang, Registration-based model reduction in complex two-dimensional geometries, J. Sci. Comput. 79 (2021) 88, http://dx.doi.org/10.1007/s10915-021-01584-y.
[62] F. Salmoiraghi, A. Scardigli, H. Telib, G. Rozza, Free-form deformation, mesh morphing and reduced-order methods: enablers for efficient aerodynamic shape optimisation, Int. J. Comput. Fluid Dyn. 34 (2018) 233–247, http://dx.doi.org/10.1080/10618562.2018.1514115.
[63] R. Soares, K. Garry, J. Holt, Comparison of the far-field aerodynamic wake development for three DrivAer model configurations using a cost-effective RANS simulation, in: SAE Technical Paper 2017-01-1514, 2017, http://dx.doi.org/10.4271/2017-01-1514.
[64] E. Papoutsis-Kiachagias, V. Asouti, K. Giannakoglou, et al., Multi-point aerodynamic shape optimization of cars based on continuous adjoint, Struct. Multidiscip. Optim. 59 (2019) 675–694, http://dx.doi.org/10.1007/s00158-018-2091-3.
[65] G. Wang, Adaptive response surface method using inherited latin hypercube design points, J. Mech. Des. 125 (2003) 210–220, http://dx.doi.org/10.1115/1.1561044.

Fausto Dicech holds a Master Degree in Mechanical Engineering since 2022 from the University of Trieste, where also he is now a second year Ph.D. student in Industrial and Information Engineering. Concurrently, he is collaborating in the research activities of Optimad srl. His research and topics of interest focus on multi-fidelity techniques, reduced order modeling and optimization.

Konstantinos Gkaragkounis has been a numerical methods developer at Optimad srl since 2021, focusing on shape optimization and geometry manipulation methods. Before that, at NTUA/PCOpt, he completed his Ph.D. titled "The Continuous Adjoint Method in Aerodynamic and Conjugate Heat Transfer Shape Optimization, for Turbulent Flows" and worked as a researcher on topology optimization in conjugate heat transfer problems for industry.

Lucia Parussini received her Doctorate in Chemical and Energetic Technologies from the University of Trieste in 2007. After a few years as post-doctoral fellow in Fluid Machinery at the University of Trieste, she has been Quantitative Analyst Manager in modefinance srl till 2012, when she got the Assistant Professor position in Fluid Machinery at University of Trieste. Since 2020 she is Associate Professor. She is currently Lecturer of Fluid Dynamics, Fluid Machines and Energy Systems, Mechanical Thermal Measurements and Testing in the Department of Engineering and Architecture. Her research interests are: Methods for the resolution of thermo-fluid dynamic problems; Uncertainty Quantification; Optimization; Response surfaces.

Anna Spagnolo graduated with a Master's Degree in Mathematics from the University of Trieste in 2022, focusing her studies on computational modeling and optimization. She has been a software developer in Optimad srl since 2022, dealing with machine learning techniques such as multi-fidelity and reduced order models.

Haysam Telib is the CEO of Optimad srl since 2009. In 2006 he obtained a joint Ph.D. degree in Fluid Mechanics from the Politecnico di Torino and the École Centrale de Lyon. For the next three years he was in Politecnico di Torino as a Fellow Researcher on the topics of numerical schemes and reduced order models. His research focus on numerical methods, reduced order modeling, immersed boundary and computational fluid dynamics.