An efficient computational framework for naval shape design and optimization problems by means of data-driven reduced order modeling techniques

Nicola Demo[1] · Giulio Ortali[1,2] · Gianluca Gustin[3] · Gianluigi Rozza[1] ·
Gianpiero Lavini[3]

Abstract

This contribution describes the implementation of a data-driven shape optimization pipeline in a naval architecture application. We adopt reduced order models in order to improve the efficiency of the overall optimization, keeping a modular and equation-free nature to target the industrial demand. We applied the above mentioned pipeline to a realistic cruise ship in order to reduce the total drag. We begin by defining the design space, generated by deforming an initial shape in a parametric way using free form deformation. The evaluation of the performance of each new hull is determined by simulating the flux via finite volume discretization of a two-phase (water and air) fluid. Since the fluid dynamics model can result very expensive—especially dealing with complex industrial geometries—we propose also a dynamic mode decomposition enhancement to reduce the computational cost of a single numerical simulation. The real-time computation is finally achieved by means of proper orthogonal decomposition with Gaussian process regression technique. Thanks to the quick approximation, a genetic optimization algorithm becomes feasible to converge towards the optimal shape.

✉ Gianluigi Rozza
gianluigi.rozza@sissa.it

Nicola Demo
nicola.demo@sissa.it

Giulio Ortali
giulio.ortali@sissa.it

Gianluca Gustin
gianluca.gustin@fincantieri.it

Gianpiero Lavini
gianpiero.lavini@fincantieri.it

[1] Mathematics Area, mathLab, SISSA, International School of Advanced Studies, Trieste, Italy

[2] Department of Mathematical Sciences, Politecnico di Torino, Turin, Italy

[3] Fincantieri-Divisione Navi Mercantili e Passeggeri, FINCANTIERI SpA, Trieste, Italy

 Springer

1 Introduction and motivations

A shape optimization problem consists of finding the geometric configuration of an object that maximizes the performance of such object. Due to the number and the complexity of methods to integrate together—i.e. a shape parametrization algorithm, a numerical solver, an optimization procedure—, this task remains challenging even nowadays. One of the most common problems is the computational cost required to solve the mathematical model, necessary to predict the performance of the deformed object. Addressing complex phenomena, even exploiting high-performance facilities, the total computational load may make the procedure unfeasible, since the performance evaluation has to be repeated for each new deformed configuration.

In this work, we extend the computational pipeline already presented in [5], using two different reduced order modeling (ROM) approaches to address the high computational demand of optimization problems based on partial differential equations (PDEs) in parametric domains. The goal is obtaining the optimal shape of the input object—in our case, the naval hull of a cruise ship—with a reasonable demand of computational resources. For different version of this shape optimization pipeline, we suggest [6] for a POD reduction to geometrical parametrization, [1,43] as example of hyper-reduction techniques application within the optimization pipeline, [9,10,44] for constrained optimization, and [27,39] for an additional parameter space analysis by means of active subspace property. ROM provides a model simplification, bartering a slightly increased error in the model output with a remarkable reduction of the computational cost. The real-time response of such models helps to accelerate the entire optimization process. Other similar framework regarding ROM have been presented in [34,40].

In details, the two adopted ROM techniques are: (i) the data-driven proper orthogonal decomposition (POD) coupled with Gaussian process regression (GPR) for the approximation of the solution manifold for the parametric model, and (ii) the dynamic mode decomposition (DMD) algorithm to estimate the regime state of the transient fluid dynamics problem. We specify that the GPR approach has be applied in this context thanks to its capability of providing good precision even with very few snapshots, but other techniques are available in a data-driven context. Among all the possible alternatives we highlight the usage of interpolation [11,36] or the usage of neural networks [17], which tends to be very accurate but requiring larger datasets. Exploiting POD-GPR and DMD, not only we need a limited number of high-fidelity (and expensive) simulations but we are able even to reduce the computational cost of the latter. The main advantage is that the optimization procedure, which has to iterate towards the optimum, uses the reduced order model to estimate the performance of any new deformed object in a very quickly manner. An additional value of the proposed framework is the complete modularity for the data-driven nature of the ROM methods. In fact, they are based only on the output of the system, without the necessity to know the governing equations or, from a technical viewpoint, to access to the discrete operators of the problem. We propose in this work an application on the shape optimization of a cruise ship, but the pipeline can be easily modified to plug different algorithms or software. All these features make the framework especially suited for industry, thanks to the huge speedup in optimization—but also design—contexts and the natural capacity to be even coupled with commercial software.
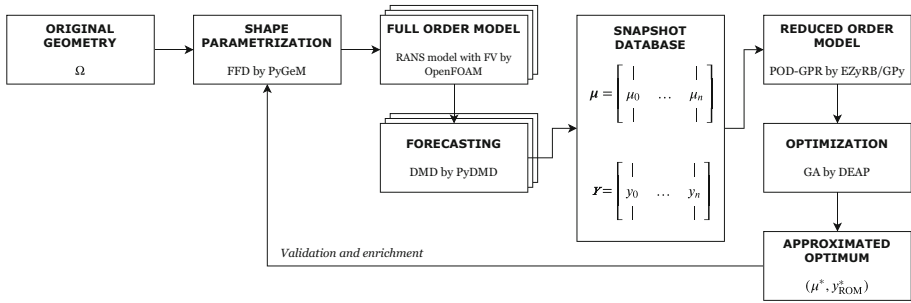
Fig. 1 The flowchart of the complete computational pipeline

The work is structured as following: in Sect. 2 we described in the details how the components are combined together, going into the deeper mathematical formulation of all of them in the next subsections. In particular: Sect. 2.1 will focus on the free-form deformation (FFD), the algorithm used for the shape parametrization; Sect. 2.2 will introduce the full-order model we adopt and its numerical solution using the finite volume (FV) approach; Sects. 2.2.1 and 2.3 will introduce the algorithmic formulation of the DMD and POD-GPR techniques, respectively; Sect. 2.4 will summarize the genetic algorithm (GA), i.e. the optimization method we used. Finally, in Sect. 3 we present the numerical setting of the resistance minimization problem for a parametric cruise ship and the results obtained by applying the described framework on it, before proposing a conclusive comment and some future perspectives in Sect. 4.

2 The complete computational pipeline

This section focuses on the integration of all the components into a single pipeline capable of optimizing an input object with a generic shape $\Omega \in \mathbb{R}^3$. We will provide details about the methodologies stack, specifying the interfaces between methods in order to let the reader capable to understand the workflow. The proposed framework can be however, thanks to the data-driven feature, easily extended, replacing one or more techniques, increasing integrability of such pipeline (Fig. 1).

As first ingredient, we need a map $\mathcal{M} : \mathbb{R}^3 \to \mathbb{R}^3$ that, depending on some numeric parameters, deforms the original domain such that $\Omega(\boldsymbol{\mu}) = \mathcal{M}(\Omega, \boldsymbol{\mu})$. Dealing with complex geometries, we chose the free-form deformation (FFD) [37] to deform the original object, because of its capability to preserve continuity on the surface derivatives and to perform global deformation even with few parameters. The parameters $\boldsymbol{\mu} \in P \subset \mathbb{R}^P$, for this method, control the displacement of some points (along some directions) belonging to a lattice of points around the object. This motion produce a deformation in all the space embedded by the lattice. Chosen the parameter space P, we sample this latter N times to obtain the set $\{\boldsymbol{\mu}_i\}_{i=1}^N$, and, using the FFD, the corresponding set of deformed shapes $\{\Omega(\boldsymbol{\mu}_i)\}_{i=1}^N$.

The performance of all the samples have to be evaluated, using an accurate numerical solver. In this case, since the analyzed problem is related to an incompressible turbulent multiphase flow, we use the Reynolds-averaged Navier–Stokes (RANS) equations with the volume of fluid (VOF) approach to describe the mathematical model, and a finite volume (FV) discretization to numerically solve it. Such model requires, both for the complex geometry and the complexity of equations, a not-negligible amount of computational resources. Even if, as in this case, the number of these high-fidelity simulations is limited to N, the overall load may

result too big. We can gain additional speedup exploiting the dynamic mode decomposition (DMD) [24] to predict the regime state of the simulation. In our test, the time-dependent problem shows a quasi-periodic behaviour, continuing to oscillate around the asymptotic configurations. DMD catches this kind of patterns in the temporal evolution of a system, allowing to easily make predictions with a good accuracy. We can combine the two techniques, by computing the initial temporal snapshots—aka the output of interest of such system at a certain time—with the high-fidelity model, then feeding the DMD algorithms with the latter in order to predict the regime snapshots. We define the snapshots $\mathbf{y}_i^k$ as the output of interest of the parametric domain $\Omega(\boldsymbol{\mu}_i)$ at time k: the regime state $\mathbf{y}_i^{m+c}$ is then predicted collecting the snapshots $\{\mathbf{y}_i^j\}_{j=0}^m$, for $i = 1, \ldots, N$. It is important to specify that the computational grids built around the objects $\Omega(\boldsymbol{\mu}_i)$ are not enforced to share the same topology, or the same number of degrees of freedom, but for the DMD is necessary that the grids do not change during the temporal evolution of the system. In this work we do not use the pressure and velocity fields as output of interest, but directly to the distribution of total resistance (over the surface of hull). Since the data-driven approach, this does not imply any additional complexity. Our database contains thus the discrete distribution of the total resistance for all the samples.

After this step, we obtain a set of N pairs composed by the input parameters and the regime states, that is $\{(\boldsymbol{\mu}_i, \mathbf{y}_i^{m+c})\}_{i=1}^N$. In case of output with different dimensions, we need to project the solution from the FV discretized space to the original deformed geometry $\Omega(\boldsymbol{\mu})$. Being originated by the FFD, all the geometries share the same topology. Assuming the geometry Ω is discretized in $\mathcal{N}$ degrees of freedom, the resulting new pairs are defined as $(\boldsymbol{\mu}_i, \hat{\mathbf{y}}_i)$, with $\boldsymbol{\mu}_i \in P$ and $\hat{\mathbf{y}}_i \in \mathbb{R}^{\mathcal{N}}$, for $i = 1, \ldots, N$.

Proper orthogonal decomposition (POD) [16] is now involved to reduce the dimensionality of the snapshots. The outputs $\hat{\mathbf{y}}_i \in \mathbb{R}^{\mathcal{N}}$ are projected onto the POD space, which typically has a very lower dimensions, obtaining the reduced space representation $\mathbf{c}_i \in \mathbb{R}^{N_{\text{POD}}}$ of the original states. The input-output pairs are now $(\boldsymbol{\mu}_i, \mathbf{c}_i)$ for $i = 1, \ldots, N$: assuming that a mapping $F : P \rightarrow \mathbb{R}^{N_{\text{POD}}}$ exists between input and output such that $\mathbf{c} = F(\boldsymbol{\mu})$, we can exploit the collected outputs to approximate the output itself for different parameter value using any interpolation/regression method. In this contribution, we adopt a Gaussian process regression (GPR) [32] to approximate the input-output relation with a Bayesian approach. Other examples for the POD-GPR coupling can be found in [15,30]. Finally, the low-dimensional output is projected back to the full-order space to obtain the approximated solution. Combining the techniques, we are able to build a reduced order model based only on the system output capable to provide an approximation of the output $\mathbf{y}_{\text{ROM}}^j$ for untried parameters $\boldsymbol{\mu}_j$ in real-time. Additionally, the uncertainty in the Gaussian process regression provides an estimator for the accuracy of the reduced order model. In our test, we remember we use the resistance distribution as output of interest.

The optimization procedure is then applied over the reduced order model, by computing the objective function on the state predicted using POD-GPR. Thanks to the negligible time required for the performance evaluation of a new shape, we can explore the parameter space with a genetic algorithm (GA) [21] to converge to the optimal shape. The quantity to minimize, in our numerical experiments, is the total resistance, that is nothing but the integral of corresponding field. The objective function relies hence on the previously mentioned methods, since to compute it we need to project the POD-GPR approximation over the new shape obtained by FFD. At the end, we get the optimal parameter $\boldsymbol{\mu}^*$ and correspondent (approximated) output $\mathbf{y}_{\text{ROM}}^*$. Such parameter can be used to restart the pipeline, performing the morphing over the geometry then testing it by using the high-fidelity solver for the valida-

tion of the result. Not only: this latter simulation can be further exploited by adding it to the snapshots database, resulting in an iterative process where the approximated output is used for the reduced order model, enriching in this way the accuracy of the model itself. Thanks to this *validation and enrichment* step, we are able to limit the error induced by the ROM techniques: due to equation-free nature of the pipeline, we are not able to bound the error without any information regarding the full-order model, but we can estimate it by validating the parametric configuration with the high-fidelity solutions.

2.1 Free-form deformation for shape parametrization

Free-form deformation (FFD) is a geometric tool, extensively employed in computer graphics, used to deform a rigid object based on the movement of some predefined control points. Introduced in [37], it has seen various improvements over the years. The reader can refer for example for a more recent review [3,25,35] for a coupling with ROM techniques. The main idea behind FFD is to define a regular lattice of points around the object (or part of it) and manipulate the whole embedded space by moving some of those control points. Mathematically, this is obtained by mapping the physical space enclosed by the lattice to a unit cube $D = [0, 1]^d$ by using an invertible map $\psi : \mathbb{R}^d \to D$.

Inside the unit cube we define a cubic lattice of control points, with L, M and N points respectively in x, y and z directions:

$$P_{l,m,n}^0 = \begin{pmatrix} l/L \\ m/M \\ n/N \end{pmatrix} \in D, \tag{1}$$

where $l = 0, \ldots, L, m = 0, \ldots, M$ and $n = 0, \ldots, N$. We move these points by adding a motion $\mu_{l,m,n}$ such that:

$$P_{l,m,n} = P_{l,m,n}^0 + \mu_{l,m,n}. \tag{2}$$

The parametric map $\hat{T} : D \to D$ that performs the deformation of reference space is then defined by:

$$\hat{T}(s, t, p; \mu) = \sum_{l=0}^{L} \sum_{m=0}^{M} \sum_{n=0}^{N} b_l^L(s) b_m^M(t) b_n^N(p) P_{l,m,n}, \tag{3}$$

where:

$$\begin{aligned} b_l^L(s) &= \binom{L}{l}(1 - s)^{(L-l)} s^l, \\ b_m^M(t) &= \binom{M}{m}(1 - t)^{(M-m)} t^m, \\ b_n^N(p) &= \binom{N}{n}(1 - p)^{(N-n)} p^n. \end{aligned} \tag{4}$$

The FFD map $T : \mathbb{R}^3 \to \mathbb{R}^3$ is then composed as it follows:

$$T(\cdot; \mu) = (\psi^{-1} \circ \hat{T} \circ \psi)(\cdot; \mu). \tag{5}$$

We applied the FFD algorithm directly to input object using the open source Python package called PyGeM [31].

2.2 Finite volume for high-fidelity database

We now discuss the full order model (FOM), which generates what we call the high fidelity solution. The Reynolds-averaged Navier–Stokes (RANS) equations model the turbulent incompressible flow around the naval hull, while for the modeling of the two different phases—e.g. water and air—we adopt the volume of fluid (VOF) technique [20]. The equations governing our system are then:

$$\begin{cases} \frac{\partial \bar{u}}{\partial t} + (\bar{u} \cdot \nabla)\bar{u} + \frac{1}{\rho}\nabla\bar{p} - \nabla \cdot \nu\nabla\bar{u} - \nabla \cdot (\tilde{u} \otimes \tilde{u}) = 0, \\ \nabla \cdot \bar{u} = 0, \\ \frac{\partial \alpha}{\partial t} + \nabla \cdot (\bar{u}\alpha) = 0, \end{cases} \tag{6}$$

where $\bar{u}$ and $\tilde{u}$ refer the mean and fluctuating velocity after the RANS decomposition, $\bar{p}$ denote the (mean) pressure, ρ is the density, ν the kinematic viscosity and α is the discontinuous variable belonging to interval [0, 1] representing the fraction of the second flow in the infinitesimal volume.

The first two equations are the continuity and momentum conservation, while the third one represent the transport equation for the VOF variable α. The Reynolds stresses tensor $\tilde{u} \otimes \tilde{u}$ can be modeled by adding additional equations in order to close the system: in this work, we use the SST$k - \omega$ turbulence model [26]. For the multiphase nature of the flow, the density ρ and the kinematic viscosity ν are defined using an algebraic formula expressing them as a convex combination of the corresponding properties of the two flows:

$$\begin{aligned} \rho &= \alpha\rho_1 + (1 - \alpha)\rho_2, \\ \nu &= \alpha\nu_1 + (1 - \alpha)\nu_2. \end{aligned} \tag{7}$$

To solve such problem, we apply the finite volume (FV) approach. We adopted a 1st order implicit Euler scheme for the temporal discretization, while for the spatial scheme we apply the linear upwind one. Regarding the software, the simulation is carried out using the C++ library OpenFOAM [28].

2.2.1 Dynamic mode decomposition for regime state prediction

Dynamic mode decomposition (DMD) is a data-driven ROM technique that approximates the evolution of a complex dynamical system as the combination of few features linearly evolving in time [24,36]. The basic idea is to provide a low-dimensional approximation of the Koopman operator [23] based on few temporarily equispaced snapshots of the studied system. DMD assumes the evolution of the latter can be expressed as:

$$\mathbf{y}_{k+1} = \mathbf{A}\mathbf{y}_k, \tag{8}$$

where $\mathbf{y}_{k+1} \in \mathbb{R}^{\mathcal{N}}$ and $\mathbf{y}_k \in \mathbb{R}^{\mathcal{N}}$ are two snapshots at the time $t = k$ and $t = k + 1$, respectively, while $\mathbf{A}$ refers to a discrete linear operator. A least-square approach can be used to calculate this operator. After collecting a set of snapshots defined as $\{\mathbf{y}_{t_0+k\Delta t}\}_{k=0}^{M}$, we can arrange them into two matrices $\mathbf{Y} = \begin{bmatrix} \mathbf{y}_0 & \dots & \mathbf{y}_{M-1} \end{bmatrix}, \dot{\mathbf{Y}} = \begin{bmatrix} \mathbf{y}_1 & \dots & \mathbf{y}_M \end{bmatrix} \in \mathbb{R}^{\mathcal{N} \times M}$ such that the correspondent columns of the two matrices represent two sequential snapshots.

We can now minimizing the error $\|\mathbf{A}\mathbf{Y} - \dot{\mathbf{Y}}\|_F$ by the following matrix multiplication $\mathbf{A} = \dot{\mathbf{Y}}\mathbf{Y}^{\dagger}$, where the symbol † indicates the Moore–Penrose pseudoinverse. While we can already use the operator $\mathbf{A}$ to analyze the system, in practice because of its considerable dimension and the difficulties that would arise in order to obtain it numerically. DMD uses

then the singular value decomposition (SVD) to compute the reduced space onto which projecting the operator. Formally

$$\mathbf{Y} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T, \tag{9}$$

where $\mathbf{U} \in \mathbb{R}^{\mathcal{N} \times M}$, $\mathbf{\Sigma} \in \mathbb{R}^{M \times M}$ and $\mathbf{V} \in \mathbb{R}^{M \times M}$. The left singular vectors (the columns of $\mathbf{U}$) span the optimal low-dimensional space, allowing us to project the operator $\mathbf{A}$ onto it:

$$\tilde{\mathbf{A}} = U^T \mathbf{A} \mathbf{U} = \mathbf{U}^T \dot{\mathbf{Y}} V \Sigma^{-1} \tag{10}$$

to compute the reduced operator. The interesting feature is that the eigenvalues of $\tilde{\mathbf{A}}$ are equal to the non-zero ones of the high dimensional operator $\mathbf{A}$, and also the eigenvectors of the two operators are related each other [41]. In particular:

$$\mathbf{\Phi} = \dot{\mathbf{Y}}\mathbf{V}\mathbf{\Sigma}^{-1}\mathbf{W}, \tag{11}$$

where Φ is the matrix containing the $\mathbf{A}$ eigenvectors, the so-called DMD modes, and $\mathbf{W}$ is the matrix of $\tilde{\mathbf{A}}$ eigenvectors. Defining $\mathbf{\Lambda}$ as the diagonal matrix of eigenvalues, we have:

$$\mathbf{Y} \approx \mathbf{A}\mathbf{X} = \mathbf{\Phi}\mathbf{\Lambda}\mathbf{\Phi}^{\dagger}\mathbf{X} \tag{12}$$

that implies that any snapshots can be approximated computing $\mathbf{y}_k = \mathbf{\Phi}\mathbf{\Lambda}^k\mathbf{\Phi}^{\dagger}\mathbf{y}_0$.

We apply the DMD on the snapshots coming from the full-order model (discussed in Sect. 2.2) in order to perform fewer temporal iterations using the high-fidelity solver, and predict the output we are interested to analyze in order to gain an additional considerable speedup. The results are obtained using PyDMD [8], a Python package that implements the most common version of DMD.

2.3 Reduced order model exploiting proper orthogonal decomposition

Reduced basis (RB) is a ROM method that approximates the solution manifold of a parametric problem using a low number of basis functions that form what we call the reduced basis [16, 33]. In this community, proper orthogonal decomposition (POD) is a widespread technique [4,38] since its capability to provide orthogonal basis that have an energetically hierarchy. While a possible approach for turbulent flows involving projection-based ROM is available in [18], we prefer the data-driven approach for the higher integrability in many industrial workflows. POD needs as input a matrix containing samples of the solution manifold. We define $\mathcal{N}$ the number of degrees of freedom of our numerical model and $\mathbf{y} \in \mathbb{R}^{\mathcal{N}}$ its solution for a generic parameter $\boldsymbol{\mu}$. Thus, the snapshots matrix $\mathbf{Y} \in \mathbb{R}^{\mathcal{N} \times n}$ is defined as:

$$\mathbf{Y} = \begin{bmatrix} | & | & & | \\ \mathbf{y}_1 & \mathbf{y}_2 & \cdots & \mathbf{y}_n \\ | & | & & | \end{bmatrix}. \tag{13}$$

The POD basis is defined as basis that maximizes the similarity (as measured by the square of the scalar product) between the snapshots matrix and its elements, under the constraint of orthonormality. Formally, the POD basis $\{\psi_i\}_{i=0}^{l}$ of dimension l is defined as:

$$\max_{\psi_1,\dots\psi_l} \sum_{i=1}^{l} \sum_{j=1}^{n} |\langle \mathbf{y}_j, \psi_i \rangle_{\mathbb{R}^{\mathcal{N}}}|^2 \tag{14}$$

such that $\langle \tilde{\psi}_i, \tilde{\psi}_i \rangle_{\mathbb{R}^{\mathcal{N}}} = \delta_{i,j}$, for $1 \leq i, j \leq l$. Singular value decomposition (SVD) is a method that computes the POD basis [42] by decomposing the snapshots matrix:

$$\mathbf{Y} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^*, \tag{15}$$

where matrices $\mathbf{U} \in \mathbb{R}^{\mathcal{N} \times n}$ and $\mathbf{V} \in \mathbb{R}^{n \times n}$ are unitary while $\mathbf{\Sigma} \in \mathbb{R}^{n \times n}$ is diagonal. In particular, the columns of $\mathbf{U}$ are POD basis. We project the original snapshots onto the POD space to have a low-dimensional representation. In matrix form:

$$\mathbf{C} = \mathbf{U}^T \mathbf{Y}, \quad \mathbf{c} \in \mathbb{R}^{n \times n}. \tag{16}$$

The columns of $\mathbf{C}$ are the modal coefficients $\mathbf{c}_i \in \mathbb{R}^n$.

We can now exploit this reduced space in order to build a probabilistic response surface using the Gaussian process regression (GPR) [32]. In particular, assuming there is a natural relation $F : P \to \mathbb{R}^n$ between our geometric parameters $\boldsymbol{\mu}$ and the low-dimensional output $\mathbf{c}$ such that $\mathbf{c} = F(\boldsymbol{\mu})$, we try to approximate it with a multivariate Gaussian distribution. We define:

$$f(\boldsymbol{\mu}) \sim \text{GP}(\mathcal{M}(\boldsymbol{\mu}), \mathcal{K}(\boldsymbol{\mu}, \boldsymbol{\mu})), \tag{17}$$

where $\mathcal{M}$ refers to the mean of the distribution and $\mathcal{K}$ to its covariance. There are many possible choices for the covariance function $\mathcal{K} : P \times P \to \mathbb{R}$, in our case we use the squared exponential one defined as $\mathcal{K}_{SE}(\boldsymbol{\mu}_i, \boldsymbol{\mu}_j) = \sigma^2 \exp(-\frac{1}{2}\|\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\|^2)$. The prior joint Gaussian distribution for the outputs $\mathbf{c}$ results then

$$\mathbf{c}|\boldsymbol{\mu} \sim \mathcal{N}(0, \mathcal{K}(\boldsymbol{\mu}, \boldsymbol{\mu})). \tag{18}$$

Being the output a vector $\mathbf{c} = \begin{bmatrix} c_1 & \ldots & c_n \end{bmatrix}$, we apply the GPR to each component, treating these components as independent variable. For sake of simplicity we assume that the GP has mean equal to zero: the entire process results defined only by the covariance function. In order to specify the GP for our dataset, we need to maximize the marginal likelihood varying the hyper-parameters of the covariance function, in this case only the σ. Once obtained the output distribution, we can just sample it at the test parameters to predict the output—which, we remember, is the low-dimensional snapshot—by exploiting the joint distribution:

$$\bar{\mathbf{c}}|\bar{\boldsymbol{\mu}}, \boldsymbol{\mu}, \mathbf{c} \sim \mathcal{N}(\mathbf{m}, \mathbf{C}) \tag{19}$$

with

$$\begin{aligned} \mathbf{m} &= \mathcal{K}(\bar{\boldsymbol{\mu}}, \boldsymbol{\mu})\mathcal{K}(\boldsymbol{\mu}, \boldsymbol{\mu})^{-1}\mathbf{c}, \\ \mathbf{C} &= \mathcal{K}(\bar{\boldsymbol{\mu}}, \bar{\boldsymbol{\mu}}) - \mathcal{K}(\bar{\boldsymbol{\mu}}, \boldsymbol{\mu})\mathcal{K}(\boldsymbol{\mu}, \boldsymbol{\mu})^{-1}\mathcal{K}(\boldsymbol{\mu}, \bar{\boldsymbol{\mu}}), \end{aligned} \tag{20}$$

where $\boldsymbol{\mu}$ and $\bar{\boldsymbol{\mu}}$ refer to the input parameters and the test parameters, and where $\mathbf{c}$ and $\bar{\mathbf{c}}$ are the corresponding train and test output.

We compute the modal coefficients of all (untested) new parameters. To approximate the high-dimensional snapshots we need just to back map the modal coefficients to the original space. In matricial form:

$$\bar{\mathbf{y}} = \mathbf{U}\bar{\mathbf{c}}. \tag{21}$$

An additional gain of such method is the complete division between two computational phases often called *offline* and *online* steps We can easily note that, to collect the input snapshots, we initially need to compute several snapshots using the chosen high-fidelity model. This is the most expensive part, and usually is carried out on powerful machines. This offline step is fortunately independent from the online one, where actually the snapshots are

combined to span the reduced space and approximate the new reduced snapshots. Since this latter can be easily performed on standard laptops, the computational splitting in two steps allows also to efficiently exploit all different resources.

For the implementation, we developed and released this version of data-driven POD in the numerical open source package EZyRB [7], exploiting the library GPy [14] for the GPR step.

2.4 Genetic algorithm for global optimization

Genetic algorithms (GA) denote in literature the family of computational methods that are inspired by Darwin's theory of evolution. In an optimization context, emulating the natural behaviour of living beings, this methodology gained popularity due to its easy application and the capability to not get blocked in local minima. The algorithm was initially proposed by Holland in [2,21,22] and it is based on few fundamental steps: selection, mutation and mate. We consider any sample of the parameter domain as an individual $\boldsymbol{\mu}_i \in P \subset \mathbb{R}^P$ with P chromosomes. The fitness of the individuals is quantify by a scalar objective function $f : P \rightarrow \mathbb{R}$. We define the initial population $\mathbf{M}^0 = \{\boldsymbol{\mu}_i\}_{i=1}^{N_0}$ composed by N_0 individuals that are randomly created within the parameter space. The corresponding fitnesses are compute and the evolutive process of individuals starts.

The first step is the *selection* of the best individuals in the population. Intuitively, the basic approach results choosing the N individuals that have the highest fitnesses, but for large population, or simply to reinforce the stochastic component of the method, a probabilistic selection can be performed. The selected individuals are often referred as the offspring that will breed the future generation.

We are now ready to reproduce the random evolution of such individuals. This is done in the *mutation* and *mate* steps. In the mutation, chromosomes of the individuals can change, partially or entirely, in order to create the new individuals. Several approaches are available for the mutation, but usually they are based to a mutation probability to reproduce the aleatory nature of evolution. In the mate step, individuals are coupled into pairs and, still randomly, the chromosomes of the parent individuals are combined to originate the two children. In particular, the mate step emulates the reproduction step, and for this reason can be usually called also cross-over.

The population is now composed by the new (mated and mutated) individuals. Iterating this process, the population will converge toward the optimal individual, but depending on the shape of fitness function it may requires many generations to converge.

For the numerical experiments, we use the GA implementation provided by the DEAP [13] package, an open source library for evolutionary algorithms.

3 Numerical results: a cruise ship shape optimization

In this section, we will present the results obtained by applying the described computational pipeline to optimize the shape of a cruise ship. We maintain the same structure of the previous section, discussing the intermediate results for any mentioned technique (Fig. 2).

Free-form deformation We set the domain D, aka the space enclosed by the lattice of FFD control points, in order to deform only the immersed part of the hull, in the proximity of the bow. The lattice is illustrated in Fig. 3, and we can see that it is positioned, in x direction,
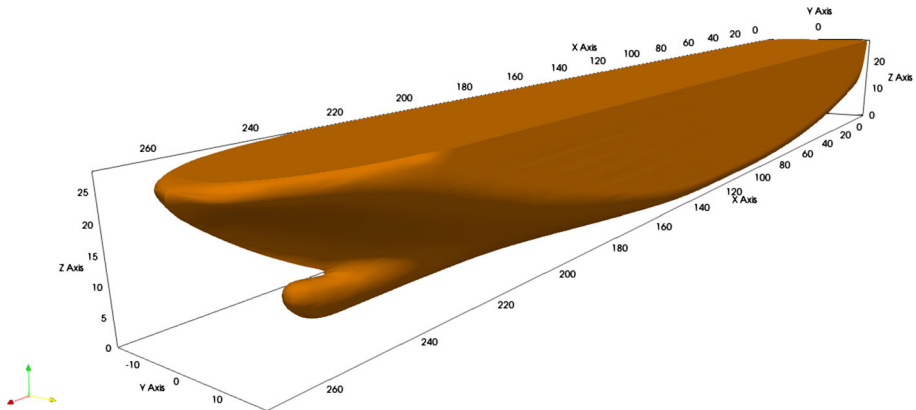
Fig. 2 The undeformed hull

on sections 10, 12, 14, 16, 18, 20 and 22.[1] For z direction, the points are displaced around the waterline, while along y axis the points are positioned for the entire width of the ship. In total, 539 FFD points are used.

Concerning the motion of such points only part of the points in the lattice are displaced: we use 6 parameters to control the movements along x (the first three parameters) and along y. An example of this motion is sketched in Fig. 4, where red arrows refers to control points movements. The layers corresponding to sections 10, 12, 20 and 22 remain fixed, together with the two upper and lower layers, the two far left and the two far-right layers and, finally, the layer over the longitudinal symmetry plane. Except for this last one, that is kept fixed to maintain symmetry, the other layers are kept fixed in order to achieve the continuity and smoothness of the shapes, required especially in the x direction where the deformation must link in a smooth way to the rest of the boat.

The parameter range have been chosen in order to avoid a high decrease of the hull volume and, at the same time, explores a large variety of new shapes. In details, we have a tolerance of the $1^0/_{00}$ for the volume constraint. With a trial and error approach we define the parameter ranges, obtaining a parameter space that is $P = [-0.08, 0.08]^6$ (the dimension of such space is the number of parameters, 6 in this test). We underline that the parameters refer to the motion normalized for the D length along the corresponding direction.

We create a set of 100 samples taking with uniform distribution on the parametric space. These are the input parameters of the high-fidelity database required for ROM.

Finite volume discretization We simulate the flow pasting around the ship using the FV method, computing for each deformed object the distribution of the total resistance over the hull. The simulations are run on model scale (1:25). The computational grid (defined in $[-39, 24] \times [-29, 0] \times [-24, 6]$) is built from scratch around all the deformed hulls, enforcing the mesh quality. The computational grid counts $\approx 1.5 \times 10^6$ cells. To the VOF model, we need an extra refinement around the waterline in order to avoid a diffusive behavior of the fraction variable α, which is discontinuous. A region of the computation grid is reported in Fig. 5 for demonstrative purpose. The numerical schemes adopted are mentioned in Sect. 2.2, and we report in Table 1 the main physical quantities we fix in our setting. The Reynolds number is near to 2×10^7. The integration in time is carried out for $t \in [0, 40]$ s, with an

---

[1] In naval architecture a boat is divided, no matter the size, in 20 chunks, generated by 21 equally spaced cuts obtained with planes perpendicular to the x-axis.
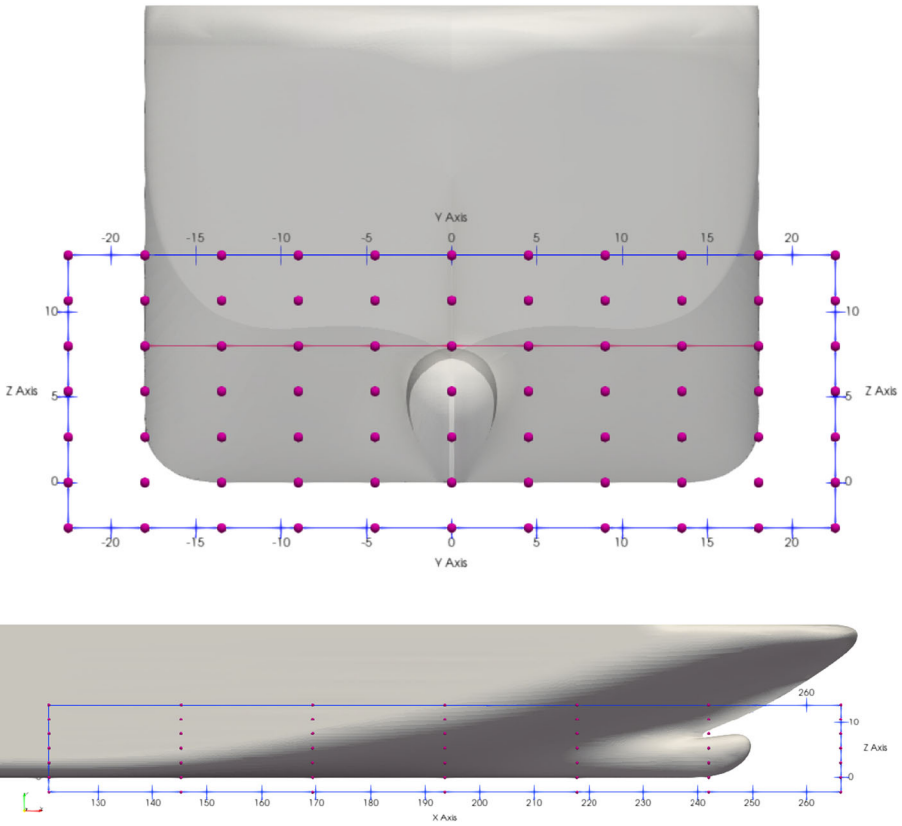
Fig. 3 x- and y-normal view of the set D (in blue) and the lattice of points $P^0_{l,m,n}$ (in red) over the undeformed hull (color figure online)
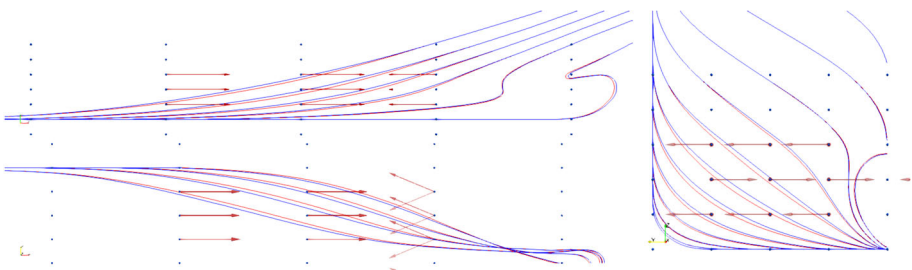
Fig. 4 Example of shape morphing with $\boldsymbol{\mu} = [0.08, 0.08, -0.06, 0.08, -0.08, 0.08]$

initial step of $\Delta t = 1 \times 10^{-3}$ s and an adjustable time-stepping governed by the Courant number (we impose it to be lower than 5). We clarify that, even if the time stepping is not fixed, we save the equispaced temporal snapshots of the system in order to feed the DMD algorithm. In this work, we are interested to the total resistance of the ship: after computing the pressure, velocity and fraction variable unknowns (from the VOF-RANS model), we can exploit them in order to calculate the resistance distribution (both the viscous and the friction
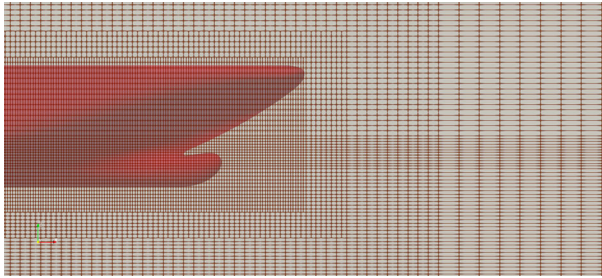
Fig. 5 The refined computational grid

Table 1 Summary of the numerical variable for VOF-RANS model

Inlet velocity u	2.263 36 m/s
Water density ρ_W	$1.09 \times 10^{-6}\,\mathrm{m^2/s}$
Air density ρ_A	$1.09 \times 10^{-5}\,\mathrm{m^2/s}$
Water kinematic viscosity ν_W	998.8 kg/m^3
Air kinematic viscosity ν_A	1 kg/m^3
Dissipation rate ω	70.497 21/s
Turbulent kinetic energy k	$7.6841 \times 10^{-4}\mathrm{m^2/s^2}$

terms) over the hull surface. Regarding the computing time, on a parallel architecture with 40 processes, the simulation lasts approximately 8 h.

Dynamic mode decomposition We applied DMD on the FOM snapshots. It is important to specify that we fit a DMD model for each geometric deformation, as a sort of post-processing on the output. We train the model using 40 snapshots per simulation, collected within the interval $t \in [20, 40)$ s with $\Delta t = 0.5$ s. The first 20 s of the simulation are discarded since they are not particularly meaningful for the boundary conditions propagation. In this contribution, we analyze the DMD operator from a spectral perspectives. Figure 6 reports in fact the eigenvalues (computed for a single simulation) after projecting the output onto a POD space of dimension 5, that is sufficient to extract the most important contribution, from an energetic perspective, of the dynamical systems. The position of eigenvalues in the complex plane provide information about the dynamics of all the DMD modes. In particular, the imaginary part is related to frequency, while the distance between them and the unit circle is related to the growth-rate. We can neglect the dumped modes (the two eigenvalues inside the circle) since their contribution is useless for future dynamics and focus on the remaining ones: two modes present a stable oscillatory trend, that actually catch the asymptotic oscillations of the FOM, and the last one $(1 + 0i)$ is practically constant. We isolate the contribution of only this latter mode, assuming it represents the regime state to which the FOM converges, using it as final output. In our setting, having built the computational grid for all the deformed ships from scratch, we need as last step to project the resistance distribution over the initial geometry $\Omega(\boldsymbol{\mu})$, in order to ensure same dimensionality for all the outputs. In our case we use a closest neighbors interpolation. Thanks to the application of DMD, we can perform fewer time iterations in the full-order model: instead of simulating 60 seconds using the full-order model, we simulate only 40 s and we exploit the collected high-fidelity snapshots for approximating the regime state with DMD. This of course implies a reduction of $^1/_3$ of the overall time required to run all the simulations.
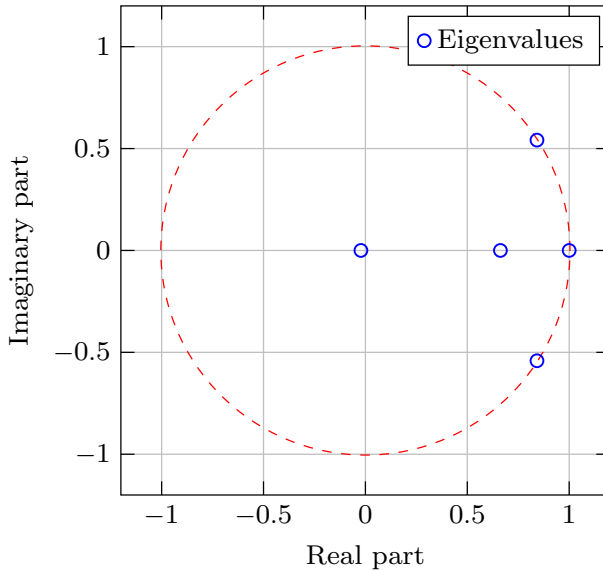
Fig. 6 First 5 DMD eigenvalues for a single simulation

Proper orthogonal decomposition with Gaussian process regression We exploit the collected database in order to build a kind of probabilistic response surface to predict the resistance of new shapes. We remember the starting set is composed by 100 input-output pairs $\{(\boldsymbol{\mu}_i, \mathbf{y}_i)\}_{i=1}^{100}$, where $\boldsymbol{\mu}$ is the geometrical parameters provided to FFD and $\mathbf{y}$ is the resistance distribution over the deformed hull. Of the entire set, we use the 80% for train the POD-GPR framework and exploit the remaining pairs to test our method. Firstly we applied the POD on the snapshots matrix to reduce the dimension of the output. In this case the singular values extracted are reported in Fig. 7 from an energetic perspective. The decay-rate is not very steep, probably due to the discontinuous component for the VOF variable α, which is directly involved in the resistance computation. Despite this, the POD allows to remarkably reduce the dimension of the output, simplifying the next phase. We exploit the computed modal coefficient in order to optimize the GP, then query for the new parametric solutions. To measure the accuracy, we propose in Figs. 8 and 9 two different sensitivity analysis varying the number of POD modes used to span the reduced space and the number of snapshots to train the method, respectively. For sake of completeness, we compare the results with similar data-driven methodologies that involve, instead the GPR, other interpolation techniques for modal coefficient approximation, as the linear interpolation one or the radial basis function (RBF) one. We propose here the simplest RBF interpolation, but we make the reader aware that better results can be achieved tuning the smoothness of RBF, producing a non-interpolating RBF method. For more details we refer [29]. The error refers to the mean relative error computed on the test dataset (of dimension 20), using the resistance distribution coming from DMD as truth solution. The GPR method is able to reach the minimum error respect to the other interpolation, resulting in a relative error near to 5% adopting 20 modes, but reducing its accuracy increasing the number of modes. This trend is shown also by RBF error, that after an initial decreasing, becomes very large for many modes. Using a higher number of modes we can, in fact, increase the precision of projecting the solutions onto the
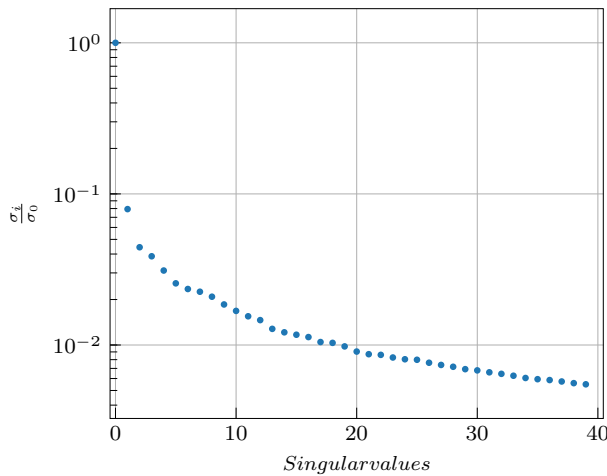
Fig. 7 Values of $\frac{\sigma_i}{\sigma_0}$ for the first 40 singular values, ordered from the highest to the lowest value

reduced space, but—since we apply all the methods using a component-wise approach—this implies an increasing number of interpolation/regression. The gain of having more dimension in the reduced space is vanished by the sum of all the interpolation errors along such dimensions. Even if also the probabilistic framework suffers this weakness, the GPR return better accuracy with respect to the RBF method. Varying the number of snapshots (Fig. 9), the difference between RBF and GPR is even more evident. While the RBF reaches an error slightly less than the 8%, the GPR is able to stay beyond the 6% with 80 snapshots. We note that we get the highest difference between the methods using few snapshots: the GPR shows higher accuracy even with few samples and a pretty constant trend for database with greater dimension. Finally, we conclude with a graphical visualization of the resistance distribution on (a limited region[2] of) the hull in Fig. 10, comparing the ROM approximation with the FOM validation. Even if the difference is notable, the reduced model can express the main physics behaviour of the original model. Regarding the computational cost reduction, the reduced model can approximate the parametric solution only sampling an already defined distribution, and even on a personal laptop it takes no more than few seconds, whereas the FV solver takes 8 h, resulting in a very huge speedup.

Genetic algorithm The goal of the entire pipeline is the minimization of the total resistance (only in the direction of the flow). To ensure feasibility of the deformed shape from the engineering viewpoint, we add a penalization on the hulls whose volume is lower that $999^0/_{00}$ of the original hull. In other words, we penalize the configurations that lead a volume decrease greater than $^1/_{1000}$ with respect to the original volume. Our optimization problem reads:

$$\min_{\boldsymbol{\mu}} \begin{cases} \int_{\Omega(\boldsymbol{\mu})} \tau_x \rho - p n_x & \text{if } \int_{\Omega(\boldsymbol{\mu})} \rho g h \geq 0.999 \int_{\Omega} \rho g h \\ \infty & \text{otherwise} \end{cases} \tag{22}$$

where τ_x is the x component of the (viscous and turbulent) tangential stresses, ρ is the density of the fluid (computed according to the VOF model), p is the pressure, n_x is the x component of the normal to the surface, g is the gravity acceleration and h is the distance between the surface and the waterline ($\int_{\Omega} \rho g h$ results the volume of the immersed hull using an

---

[2] The bulbous bow is one of the region where the pressure resistance is higher, and then difficult to predict.
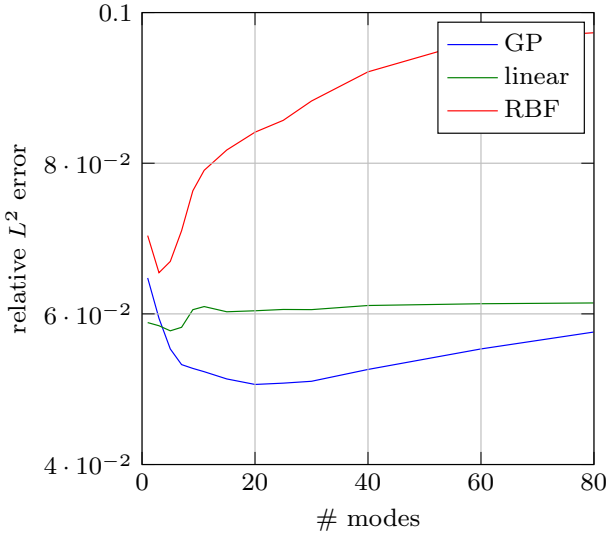
Fig. 8 Sensitivity analysis on the accuracy of POD-GPR method varying the number of POD modes used. The number of snapshots is fixed to 80
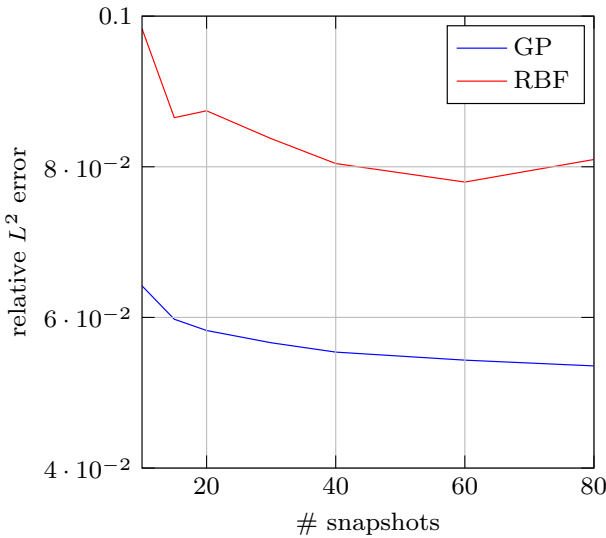
Fig. 9 Sensitivity analysis on the accuracy of POD-GPR method varying the number of snapshots. The number of POD modes is fixed to 20

hydrostatic approach). To compute the objective function for a generic parameters, we need to perform the FFD morphing then project the POD-GPR solutions over the deformed ship, in order to numerically compute such integral. We clarify that with the reduced order model returns the distribution of viscous and pressure forces over the hull, that is $\tau \rho - p$ in $\Omega(\boldsymbol{\mu})$. As already mentioned, these methods have a negligible computational cost, allowing us to optimize the shape in a very efficient manner. Despite its easiness of application, GA requires a good tuning of the hyper-parameters to result successful. In this work, we applied the one
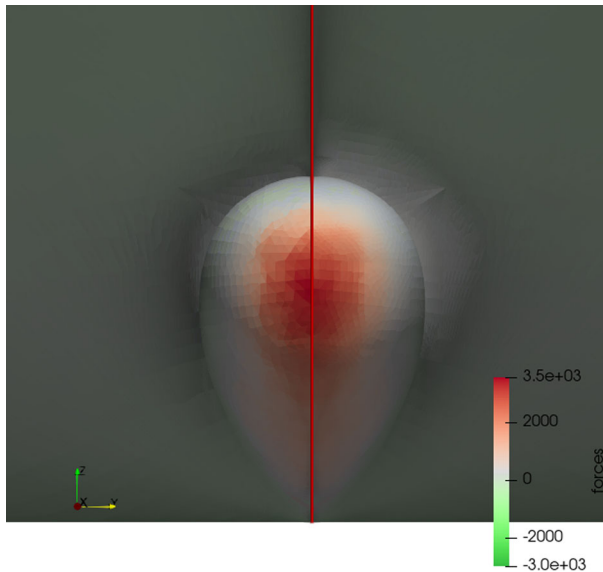
Fig. 10 Value of total resistance over the bulbous bow for the FOM (on the left) and for the ROM (on the right)

point crossover [12] for the mate procedure, while for the mutation a Gaussian mutation [19] with $\sigma = 0.1$ has been involved. We set the mate and mutation probability to 0.8 and 0.2, respectively. Moreover, we use an initial population dimension $N_0 = 200$, reducing it to $N = 30$ during the evolution. The stopping criteria in this case is the number of generations, which is set to 15. Robustness of this setting is proved in Fig. 11, where 15 different runs have been run and, for each run, the optimal shape is plotted both in terms of resistance and volume. We can note in fact that all the runs have converged to the same fitness, despite the stochastic component of the method itself, ensuring that the hyper-parameters are set to fully explore the parameter space (and then globally converge to the optimal point). The penalization we impose avoids the creation of unfeasible deformations: the optimum of all the runs show a slight decreased volume, but within the initial tolerance, while the resistance results decreased by more than the 4%.

We specify that this is the optimum for the reduced model. In order to obtain an accurate value, the optimal parameter can be plug in the pipeline and the optimal shape is then validated using the full-order FV method. Additionally, this latter can be insert in the snapshots database and used to enrich the precision of the POD-GPR model. In our case, after the validation, the gain in term of resistance is lower with respect to the ROM approximation, but reaching the 3.3% it results in a very good outcome in the engineering context. Regarding the performance point of view, the computation of the solution using the full-order model requires 8 h, on a parallel machine, while using the reduced order model just few seconds on a standard personal computer. The different architectures deny any accurate comparison, but neglecting all the minor contribution in the pipeline and counting only the high-fidelity simulations, we can estimate the computational gain. Since the optimization algorithm produces 620 individuals $(200 + 14 \times 30)$ and we use 100 samples for the construction of the reduced order space, it is easy to note that with the introduction of the POD-GPR method we can reduce the computational cost to $^1/_6$ of the original cost. Considering 8 h for a single full-order simulation
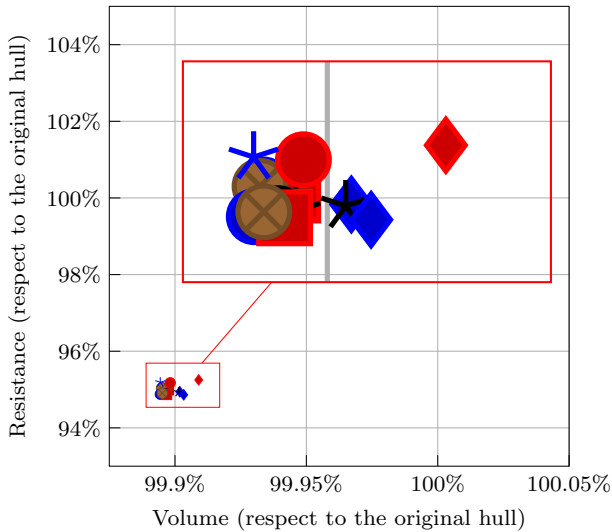
Fig. 11 Optimal shapes produced by 15 different application of GA, in terms of resistance and volume as percentage respect to the original ship

(we remark the computational gain coming by DMD is excluded in this speedup analysis, since already mentioned in previous sections), the proposed pipeline will reach the end after 800 h while replicating it without the reduction, the procedure takes 4960 h for converging. Of course, such gain grows exponentially if we repeat the optimization procedure many times: in the reduced framework we need just the 100 initial samples, independently from the number of optimization runs, while in a conventional way the 620 individuals created during the evolution have to be multiplied by the number of runs.

4 Conclusion and future perspectives

In this work, we propose a complete computational framework for shape optimization problems. To overcome the computational barrier, the dynamic mode decomposition (DMD) and the proper orthogonal decomposition with Gaussian process regression (POD-GPR) are involved. This pipeline aims at reducing the number of high-fidelity simulation needed to converge to the optimal shape, making its application very useful in all the context where the performance evaluation of the studied object results computationally expensive. We applied such framework to an industrial shape optimization problem, minimizing the total resistance of a cruise ship advancing in calm water. Exploitation of ROM techniques drastically reduces the overall time, and even if the accuracy of the reduced model is decreased (with respect to the full-order one) the final outcome presents a remarkable reduction of the resistance (3.3%).

Future developments regarding this integrated methodology may interest the extension to constraint optimization problems, the involvement of machine learning techniques in the optimization procedure, or a greedy approach that enriches the reduced order model by adding iteratively the approximated optimal shape.

References

1. Amsallem, D., Zahr, M., Choi, Y., Farhat, C.: Design optimization using hyper-reduced-order models. Struct. Multidiscip. Optim. **51**(4), 919–940 (2015)
2. Back, T.: Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms. Oxford University Press, Oxford (1996)
3. Ballarin, F., Manzoni, A., Rozza, G., Salsa, S.: Shape optimization by free-form deformation: existence results and numerical solution for Stokes flows. J. Sci. Comput. **60**(3), 537–563 (2014). https://doi.org/10.1007/s10915-013-9807-8
4. Ballarin, F., Rozza, G., Maday, Y.: Chap. Reduced-Order Semi-Implicit Schemes for Fluid-Structure Interaction Problems, pp. 149–167. Springer International Publishing, Berlin (2017). https://doi.org/10.1007/978-3-319-58786-8_10
5. Demo, N., Tezzele, M., Gustin, G., Lavini, G., Rozza, G.: Shape optimization by means of proper orthogonal decomposition and dynamic mode decomposition. In: Technology and Science for the Ships of the Future: Proceedings of NAV 2018: 19th International Conference on Ship & Maritime Research, pp. 212–219. IOS Press (2018). https://doi.org/10.3233/978-1-61499-870-9-212
6. Demo, N., Tezzele, M., Mola, A., Rozza, G.: A complete data-driven framework for the effcient solution of parametric shape design and optimisation in naval engineering problems. In: VIII International Conference on Computational Methods in Marine Engineering, pp. 111–121. International Center for Numerical Methods in Engineering (CIMNE) (2019)
7. Demo, N., Tezzele, M., Rozza, G.: EZyRB: Easy Reduced Basis method. J. Open Source Softw. **3**(24), 661 (2018). https://doi.org/10.21105/joss.00661
8. Demo, N., Tezzele, M., Rozza, G.: PyDMD: Python Dynamic Mode Decomposition. J. Open Source Softw. **3**(22), 530 (2018). https://doi.org/10.21105/joss.00530
9. Dihlmann, M., Haasdonk, B.: Certified nonlinear parameter optimization with reduced basis surrogate models. PAMM **13**(1), 3–6 (2013)
10. Dihlmann, M.A., Haasdonk, B.: Certified PDE-constrained parameter optimization using reduced basis surrogate models for evolution problems. Comput. Optim. Appl. **60**(3), 753–787 (2015)
11. Dolci, V., Arina, R.: Proper orthogonal decomposition as surrogate model for aerodynamic optimization. Int. J. Aerosp. Eng. **2016** (2016)
12. Eshelman, L.J., Schaffer, J.D.: Real-coded genetic algorithms and interval-schemata. In: Foundations of Genetic Algorithms, vol. 2, pp. 187–202. Elsevier, Amsterdam (1993)
13. Fortin, F.A., De Rainville, F.M., Gardner, M.A., Parizeau, M., Gagné, C.: DEAP: evolutionary algorithms made easy. J. Mach. Learn. Res. **13**, 2171–2175 (2012)
14. GPy: GPy: a Gaussian process framework in Python. http://github.com/SheffieldML/GPy (since 2012)
15. Guo, M., Hesthaven, J.S.: Reduced order modeling for nonlinear structural analysis using Gaussian process regression. Comput. Methods Appl. Mech. Eng. **341**, 807–826 (2018)

16. Hesthaven, J.S., Rozza, G., Stamm, B.: Certified Reduced Basis Methods for Parametrized Partial Differential Equations, 1st edn. Springer Briefs in Mathematics. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-22470-1

17. Hesthaven, J.S., Ubbiali, S.: Non-intrusive reduced order modeling of nonlinear problems using neural networks. J. Comput. Phys. **363**, 55–78 (2018)

18. Hijazi, S., Stabile, G., Mola, A., Rozza, G.: Data-Driven POD–Galerkin reduced order model for turbulent flows. J. Comput. Phys. **416**, 109513 (2020). https://doi.org/10.1016/j.jcp.2020.109513

19. Hinterding, R.: Gaussian mutation and self-adaption for numeric genetic algorithms. In: Proceedings of 1995 IEEE International Conference on Evolutionary Computation, vol. 1, p. 384. IEEE (1995)

20. Hirt, C., Nichols, B.: Volume of fluid (VOF) method for the dynamics of free boundaries. J. Comput. Phys. **39**, 201–225 (1981). https://doi.org/10.1016/0021-9991(81)90145-5

21. Holland, J.H.: Genetic algorithms and the optimal allocation of trials. SIAM J. Comput. **2**(2), 88–105 (1973)

22. Holland, J.H.: Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence. MIT Press, Cambridge (1992)

23. Koopman, B.O.: Hamiltonian systems and transformation in Hilbert space. Proc. Natl. Acad. Sci. USA **17**(5), 315 (1931)

24. Kutz, J.N., Brunton, S.L., Brunton, B.W., Proctor, J.L.: Dynamic Mode Decomposition: Data-Driven Modeling of Complex Systems. SIAM, Philadelphia (2016)

25. Lassila, T., Rozza, G.: Parametric free-form shape design with PDE models and reduced basis method. Comput. Methods Appl. Mech. Eng. **199**(23–24), 1583–1592 (2010). https://doi.org/10.1016/j.cma.2010.01.007

26. Menter, F.: Zonal two equation kw turbulence models for aerodynamic flows. In: 23rd Fluid Dynamics, Plasmadynamics, and Lasers Conference, p. 2906 (1993)

27. Mola, A., Tezzele, M., Gadalla, M., Valdenazzi, F., Grassi, D., Padovan, R., Rozza, G.: Efficient reduction in shape parameter space dimension for ship propeller blade design. In: Proceedings of MARINE 2019: VIII International Conference on Computational Methods in Marine Engineering, pp. 201–212 (2019)

28. OpenCFD: OpenFOAM—The Open Source CFD Toolbox—User's Guide, 6th edn. OpenCFD Ltd., London (2018)

29. Ortali, G.: A Data-Driven Reduced Order Optimization Approach for Cruise Ship Design. Master's thesis, Politecnico di Torino (2019). https://webthesis.biblio.polito.it/11993/. Accessed 18 Feb 2020

30. Ortali, G., Demo, N., Rozza, G., Canuto, C.: Gaussian process approach within a data-driven POD framework for fluid dynamics engineering problems (2020, submitted)

31. PyGeM: Python Geometrical Morphing. https://github.com/mathLab/PyGeM

32. Quiñonero-Candela, J., Rasmussen, C.E.: A unifying view of sparse approximate gaussian process regression. J. Mach. Learn. Res. **6**(Dec), 1939–1959 (2005)

33. Rozza, G., Hess, M.W., Stabile, G., Tezzele, M., Ballarin, F.: Basic ideas and tools for projection-based reduced order methods: preliminaries and warming-up. In: Benner, P., Grivet-Talocia, S., Quarteroni, A., Rozza, G., Schilders, W.H.A., Silveira, L.M. (eds.) Handbook on Model Order Reduction, vol. 1. De Gruyter, Berlin (2019)

34. Rozza, G., Malik, M.H., Demo, N., Tezzele, M., Girfoglio, M., Stabile, G., Mola, A.: Advances in reduced order methods for parametric industrial problems in computational fluid dynamics. In: Owen, R., de Borst, R., Reese, J., Chris, P. (eds.) ECCOMAS ECFD 7—Proceedings of 6th European Conference on Computational Mechanics (ECCM 6) and 7th European Conference on Computational Fluid Dynamics (ECFD 7), pp. 59–76. Glasgow UK (2018)

35. Salmoiraghi, F., Scardigli, A., Telib, H., Rozza, G.: Free-form deformation, mesh morphing and reduced-order methods: enablers for efficient aerodynamic shape optimisation. Int. J. Comput. Fluid Dyn. **32**(4–5), 233–247 (2018). https://doi.org/10.1080/10618562.2018.1514115

36. Schmid, P.J.: Dynamic mode decomposition of numerical and experimental data. J. Fluid Mech. **656**, 5–28 (2010)

37. Sederberg, T.W., Parry, S.R.: Free-form deformation of solid geometric models. In: Proceedings of the 13th annual conference on Computer graphics and interactive techniques, pp. 151–160 (1986)

38. Stabile, G., Rozza, G.: Finite volume POD-Galerkin stabilised reduced order methods for the parametrised incompressible Navier–Stokes equations. Comput. Fluids **173**, 273–284 (2018). https://doi.org/10.1016/j.compfluid.2018.01.035

39. Tezzele, M., Demo, N., Rozza, G.: Shape optimization through proper orthogonal decomposition with interpolation and dynamic mode decomposition enhanced by active subspaces. In: Proceedings of MARINE 2019: VIII International Conference on Computational Methods in Marine Engineering, pp. 122–133 (2019)

40. Tezzele, M., Demo, N., Stabile, G., Mola, A., Rozza, G.: Enhancing CFD predictions in shape design problems by model and parameter space reduction (2019, submitted). arXiv preprint. arXiv:2001.05237
41. Tu, J., Rowley, C., Luchtenburg, D., Brunton, S., Kutz, N.: On dynamic mode decomposition: theory and applications. J. Comput. Dyn. **1**(2), 391–421 (2014)
42. Volkwein, S.: Proper Orthogonal Decomposition: Theory and Reduced-Order Modelling. Lecture Notes. University of Konstanz, Konstanz (2012)
43. Washabaugh, K.M., Zahr, M.J., Farhat, C.: On the use of discrete nonlinear reduced-order models for the prediction of steady-state flows past parametrically deformed complex geometries. In: 54th AIAA Aerospace Sciences Meeting, p. 1814 (2016)
44. Zahr, M.J., Farhat, C.: Progressive construction of a parametric reduced-order model for PDE-constrained optimization. Int. J. Numer. Methods Eng. **102**(5), 1111–1135 (2015)