

Physics-Informed Feature-to-Feature Learning for Design-Space Dimensionality Reduction in Shape Optimisation

Shahroz Khan^{*} and Andrea Serani[†] and Matteo Diez[‡]

CNR-INM, National Research Council-Institute of Marine Engineering, Rome, Italy

Panagiotis Kaklis[§]

Department of Naval Architecture, Ocean and Marine Engineering, University of Strathclyde, Glasgow, United Kingdom.

High-dimensional parametric design problems cause optimisers and physics simulations to suffer from the curse-of-dimensionality, resulting in high computational cost. In this work, to release this computational burden, we adopted a two-step feature-to-feature learning methodology to discover a lower-dimensional latent space, based on the combination of geometry- and physics-informed principal component analysis and the active subspace method. At the first step, statistical dependencies implicit in the design parameters encode important geometric features of the underline shape. During the second step, functional features of designs are extracted in term of previously learned geometric features. Afterwards, both geometric and functional features are augmented together to create a functionally-active subspace, whose basis not only captures the geometric variance of designs but also induces the variability in the designs' physics. As the new subspace accumulates both the functional and geometric variance, therefore, it can be exploited for efficient design exploration and the construction of improved surrogate models for designs' physics prediction. The validation and experimental studies presented in this work show the beneficial effects of the current approach in comparison to a conventional single-step feature learning.

I. Introduction

INDUSTRIAL and digital evolution have leveraged the traditional design paradigm with computational tools, such as Computer-Aided Design (CAD) and physics-based simulation, whose integration creates simulation-driven optimisation (SDO) pipelines for exploration of diverse design spaces to form optimal designs. At the heart of these pipelines are CAD-based parametric techniques, which parameterise a design to create its variations during optimisation while preserving the underline structure [1, 2] and performance evaluation codes, which guide the optimisers towards the exploration of promising regions of design spaces [3, 4].

The exploration and computational efficiency of SDO for global optimum are fundamental to the intrinsic nature and dimensionality of the design space [5], whose basis is defined with design parameters of baseline shape representing the directions for shape variability. On one hand, a well-parameterised design creates a rich high-dimensional design space, which increases the possibility of finding the most optimal design but also increases the computational cost of SDO as it requires a large number of design evaluations [6]. This cost is significant especially when Quantity of Interest (QoI), which represents the design's physics or performance, is a computationally expensive metric. On the other hand, an under parameterised design creates a low-dimensional space but weakens opportunities to find a most optimal design [7]. In practice, the former approach is more favourable as at the preliminary stage designer is unaware of which design parameter efficiently optimises the design, therefore, he/she intends to create a well-defined parametric set.

To release the computational burden of SDO, data-driven meta-modelling [8] and dimensionality reduction [9] techniques have been widely used in the different fields of science and engineering. Meta-modelling methods build surrogate models of physical simulations and dimensionality reduction techniques, which also refer as a feature or manifold learning [5], create a lower-dimensional latent representation of original space. In a design context, these

^{*}Visiting Scholar, CNR-INM, National Research Council-Institute of Marine Engineering, Italy, Rome, Via di Vallerano 139, 00128; Marie Skłodowska-Curie Research Fellow, Department of Naval Architecture, Ocean and Marine Engineering, University of Strathclyde, 16 Richmond St, Glasgow G1 1XQ, United Kingdom.

[†]Research Scientist, CNR-INM Rome, Via di Vallerano 139, 00128.

[‡]Senior Research Scientist, CNR-INM Rome, Via di Vallerano 139, 00128, AIAA Member.

[§]Reader, Department of Naval Architecture, Ocean and Marine Engineering, University of Strathclyde, 16 Richmond St, Glasgow G1 1XQ, United Kingdom.

techniques are structured on the assumption that the geometric variability in design space is not the same in all directions and there are only a few inherent feature directions which materialise most improvement in the design and these inherent features can form the basis of a new lower-dimensional latent subspace [10]. In literature, such feature extraction has been achieved with (1) linear techniques to locally construct a nonlinear global manifold such as Principal Component Analysis (PCA) [9, 11], (2) kernel function with linear reduction techniques such as Kernel PCA [5, 12], or (3) with neural network-based approaches such as auto-encoders [13].

Once a lower-dimensional subspace is identified it then serves three major advantages: (1) it facilitates the high-dimensional design visualisation during the interactive process[14]; (2) it prevents the optimiser from the exhaustive exploration of high-dimensional design spaces, thereby reducing the number of QoI evaluations and allowing faster convergence towards global optimum; and (3) it allows the construction of a low-dimensional surrogate model, which learns the nonlinear and globally coupled relationship between design parameters and QoI.

However, subspace containing only geometric variability may not be the most efficient for creating the surrogate model and running the optimisation. This is because the impact of geometric variability on design's physics might not be the same [15, 16], therefore, it is essential, especially in the context of surrogate modelling, that during the feature extraction the information about QoI should be present so the latent space includes both geometric and functional variability. To tackle this problem, a physics-informed feature learning technique called Active Subspace Method (ASM) was proposed by Lukaczyk et al. [17] and Constantine [18], which learns a lower-dimensional subspace while capturing maximum variance in QoI.

However, extraction of such features requires the knowledge of the gradients, which for complex engineering problems are difficult, if not impossible, to accurately calculate. These gradients can be locally approximated using different techniques, such as finite difference method or by approximating QoI with multivariate polynomial, radial basis or Gaussian process models, but the approximation accuracy is not guaranteed especially for the high-dimensional problems. Moreover, the subspace created with ASM is solely formulated with gradients of QoI, hence it may not immediately capture the geometric variability of the original design space. As design's geometric variance and its corresponding variability in QoI is interdependent, therefore, one could first extract the geometric features and then extract the functional features.

The technique proposed in this work is formulated around a dimension-reduction pipeline that extracts the important features of a given design space first in terms of geometric variability of designs and then in terms of the variability of QoI. We refer such extraction as two-step *physics-informed feature-to-feature learning* for the generation of lower-dimensional latent space, which alleviates the computational burden for design exploration and allows the construction of the surrogate model with reduced computational cost while maintaining significant accuracy. In our approach, the first step feature extraction captures the latent directions with the highest geometric variations while ignoring the directions with low geometric variability; this extraction is equivalent to PCA. The directions with higher geometric variability then compose the basis of a subspace as a function of design parameters, which we call *Geometrically-Active Subspace* (GAS). Afterwards, another feature extraction is performed on this subspace based on ASM, which identifies the functional features based on the gradients of QoI as a function of previously explored geometric features. These functional features span the basis vectors of a *Functionally-Active Subspace* (FAS). As this subspace accumulates both geometric and functional features, therefore, can be exploited for optimisation to explore diverse design and used for computationally efficient surrogate model training. We tested our approach on DTMB 5415 naval ship model and experiments performed in this study reveal that feature-to-feature extraction improves design space's dimensionality reduction and outperforms single set-step functional feature extraction in term of accuracy surrogate modelling and optimisation results. The sequential layout of the proposed approach is illustrated in Fig. 1. The code of this pipeline is available at <https://github.com/shahrozkhhan66/PIFFL.git>.

The remainder of this paper is organised as follows: Section II gives comprehensive details on the formulation of the proposed approach. Section III discusses the test model used for the experiment and validation. The numerical results to prove the working and feasibility of the proposed technique are presented in Section IV. The concluding remarks and opportunities for future work are included in Section V.

II. Proposed Methodology

This section gives basic details of mathematical formulation and general assumptions set for the proposed approach.

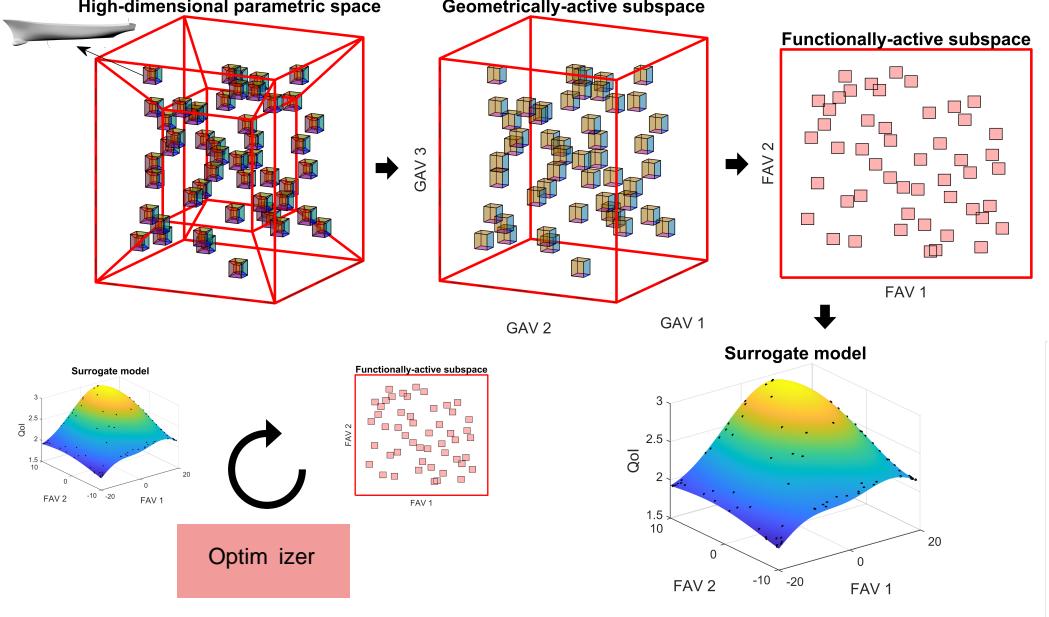


Fig. 1 Sequential workflow of the proposed approach, which commences with high-dimensional design space, represented with a hypercube, having large feasible variation of the baseline design. Afterwards, geometric features of this space are identified to span the basis of a new geometrically-active subspace. This subspace is then used to construct a functionally-active subspace to further reduced the dimensionality and to accumulate both geometric and functional variability. Thereafter, the surrogate model is developed with designs sampled from this subspace and explored with an optimiser to find an optimal design.

A. General Definitions and Assumptions

Let a geometric domain \mathcal{G} representing a baseline design and a set of coordinates $\zeta \in \mathcal{G} \subset \mathbb{R}^m$ with $m = 1, 2, 3$. Now also assume a design parameter vector $\mathbf{x} = \{x_k, k = 1, 2, \dots, M\} \in \mathcal{X} \subset \mathbb{R}^M$, where \mathcal{X} is subset of \mathbb{R}^M and is bounded with lower (\mathbf{x}^l) and upper (\mathbf{x}^u) geometric bounds creating a viable M -dimensional design space. The associated parametrisation of \mathcal{G} defines a geometric modification vector $\delta(\zeta, \mathbf{x}) \in \mathbb{R}^n$, with $n = 1, 2, 3$, which for $\mathbf{x} \in \mathcal{X}$ modifies each $\zeta \in \mathcal{G}$ of the baseline shape to new geometry $\zeta' \in \mathcal{G}'$ as:

$$\zeta' = \zeta + \delta(\zeta, \mathbf{x}), \quad (1)$$

where \mathcal{G}' is the modified representation of \mathcal{G} .

The objective here is to extract the geometric and functional features for the formulation of a reduced-dimensional representation of $\delta(\zeta, \mathbf{x})$ first in term of its geometric variability, $\hat{\delta}(\zeta, \mathbf{v})$, and then take this representation to find another reduced representation, $\hat{\delta}(\zeta, \mathbf{u})$, in terms of functional variability of \mathcal{G} with respect to QoI such as drag, resistance, stability, etc. Here, $\mathbf{v} = \{v_i, i = 1, 2, \dots, N\} \in \mathcal{V} \subset \mathbb{R}^N$ is a *Geometrically-Active Latent Variable* (GALV) vector, which is composed as a linear combination of geometrically-active features of \mathcal{X} with the shape modification vector and serves as the new dimensional coordinates for the N -dimensional GAS (\mathcal{V}). Whereas $\mathbf{u} = \{u_i, i = 1, 2, \dots, P\} \in \mathcal{U} \subset \mathbb{R}^P$ is formed as the linear combination of functionally-active features of \mathcal{V} with \mathbf{v} and define as a *Functionally-Active Latent Variable* (FALV) creating a P -dimensional FAS (\mathcal{U}), where $P < N < M$. Fig. 2 graphically illustrates shows the notation set for the formulation of the proposed approach at $m = 3, n = 3, M = 13$ and $N = 7$.

The objective here is to minimise an objective/cost function defining the reconstruction error between the original variables and their lower-dimensional representation. In present case, this error measures the accuracy of $\hat{\delta}(\zeta, \mathbf{v})$ and $\hat{\delta}(\zeta, \mathbf{u})$ in term of Mean Squared Error (MSE) normalised to total geometric (σ_g^2) and functional (σ_f^2) variance as in Eq. (2) and (3), respectively.

$$NMSE_g = \frac{MSE_g}{\sigma_g^2} = \frac{\iint_{\mathcal{X} \times \mathcal{V}, \mathcal{G}} \|\delta(\zeta, \mathbf{x}) - \hat{\delta}(\zeta, \mathbf{v})\|^2 \rho(\mathbf{x}, \mathbf{v}) d\zeta d\mathbf{x} d\mathbf{v}}{\iint_{\mathcal{X}, \mathcal{G}} \|\delta(\zeta, \mathbf{x})\|^2 \rho(\mathbf{x}) d\zeta d\mathbf{x}} \quad (2)$$

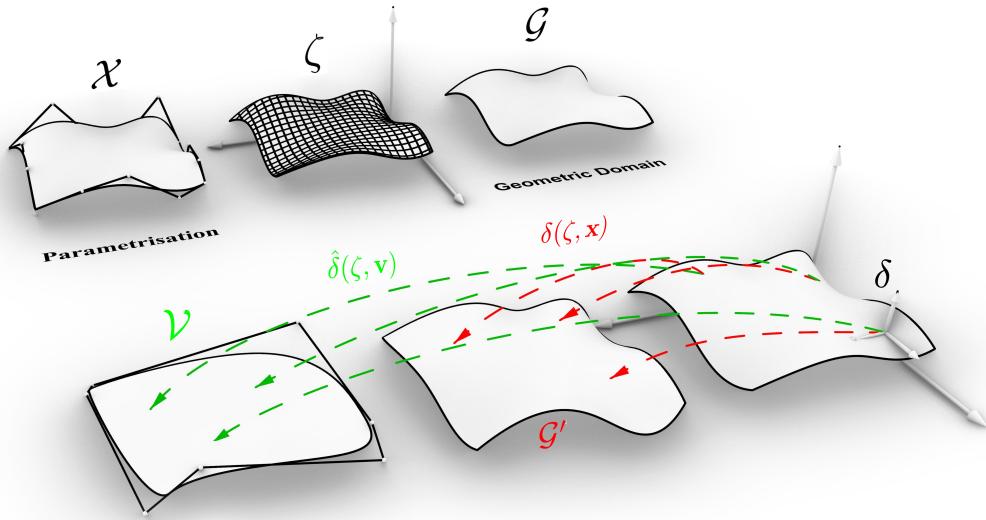


Fig. 2 Representation of the scheme and notation used for the current formulation.

$$NMSE_f = \frac{MSE_f}{\sigma_f^2} = \frac{\iint_{\mathcal{V} \times \mathcal{U}, \mathcal{G}} \|\hat{\delta}(\zeta, v) - \hat{\delta}(\zeta, u)\|^2 \rho(v, u) d\zeta dv du}{\iint_{\mathcal{V}, \mathcal{G}} \|\delta(\zeta, v)\|^2 \rho(v) d\zeta dv} \quad (3)$$

Here, $\rho(x, v)$ and $\rho(v, u)$ is a probability distribution over the product space $\mathcal{X} \times \mathcal{V}$ and $\mathcal{V} \times \mathcal{U}$, respectively.

B. Eigendecomposition

As mentioned before learning GAS and FAS is similar to the PCA and ASM, respectively, which extract features by performing the eigendecomposition of a covariance matrix, $C \in \{C^g, C^f\}$, where C^g and C^f are the covariance matrix for geometric and functional feature extraction, respectively. Once C is constructed its eigendecomposition follows the same procedure for both geometric and functional feature extraction. Therefore, we first explain the eigendecomposition and then describes the formulation of C^g and C^f in the subsequent sections.

C is a symmetric and positive definite matrix and constitutes of elements of an ambient or auxiliary space, $\mathcal{Z} \in \{\mathcal{X}, \mathcal{V}\} \subset \mathbb{R}^M$. For learning geometric features these elements are the shape associated parameters and for functional features, these elements are the gradients of QoI with respect for to the dimensional coordinates, \mathbf{z} , of \mathcal{Z} . The eigendecomposition of C for \mathcal{Z} gives:

$$C = \mathbf{W} \Lambda \mathbf{W}^T. \quad (4)$$

Herein, \mathbf{W} is the $[M \times M]$ feature matrix whose columns are orthogonal eigenvectors $\mathbf{W} = \{\mathbf{w}_k, k = 1, 2, \dots, M\}$ with $(\mathbf{w}_k^T \mathbf{w}_k = 1)$. This spans the basis of an eigenspace, which creates rotation of \mathbb{R}^M . $\Lambda = \text{diag}(\lambda_i, k = 1, 2, \dots, M)$ are the eigenvalues sorted in descending order $\lambda_1 \geq \lambda_2 \geq \dots, \lambda_M \geq 0$. To form a reduced order basis, the sorted \mathbf{W} and Λ is partitioned into two sets, containing active and inactive directions,

$$\Lambda = \begin{bmatrix} \Lambda_1 & \\ & \Lambda_2 \end{bmatrix}, \quad \mathbf{W} = \begin{bmatrix} \mathbf{W}_1 & \mathbf{W}_2 \end{bmatrix}. \quad (5)$$

The columns of $\mathbf{W}_1 = \{\mathbf{w}_i, i = 1, 2, \dots, N\}$ and $\mathbf{W}_2 = \{\mathbf{w}_j, j = 1, 2, \dots, M - N\}$ span the dominant and non-dominant features of \mathcal{Z} , which defines the active and inactive subspaces, receptively, and $\Lambda_1 = \{\lambda_i, i = 1, 2, \dots, N\}$ and $\Lambda_2 = \{\lambda_j, j = 1, 2, \dots, M - N\}$ are their corresponding eigenvalues. The separation between \mathbf{W}_1 and \mathbf{W}_2 is done based on the eigenvalues, however, they are of different nature for GAS and FAS. Therefore, the decision on number of active and inactive features for GAS and FAS is done following different theories, which will be discussed in the following sections. Here, \mathbf{W}_1 contains first N columns of \mathbf{W} corresponding to the first largest N eigenvalues, $\Lambda_1 = \{\lambda_i, i = 1, 2, \dots, N\}$.

Afterwards, \mathbf{z} can be project on these subspaces using Eq. (6) to find its active $\mathbf{a}^1 = \{a_i^1, i = 1, 2, \dots, N\}$ and inactive $\mathbf{a}^2 = \{a_j^2, j = 1, 2, \dots, M - N\}$ latent parameters.

$$\mathbf{a}^1 = \mathbf{W}_1^T \mathbf{z} \in \mathbb{R}^N, \quad \mathbf{a}^2 = \mathbf{W}_2^T \mathbf{z} \in \mathbb{R}^{M-N}. \quad (6)$$

Among \mathbf{a}^1 and \mathbf{a}^2 , we are only interested in \mathbf{a}^1 as its basis \mathbf{W}_1 covers largest variability of either geometry or QoI, which is negligibly influenced by \mathbf{a}^2 . Here, i^{th} (j^{th}) active (inactive) variable is the linear combination of elements of i^{th} (j^{th}) column \mathbf{w}_i (\mathbf{w}_j) of \mathbf{W}_1 (\mathbf{W}_2) and coordinates of original design space, $a_i^1 = w_{1,i} z_1 + w_{2,i} z_2 + \dots + w_{M,i} z_M$ ($a_j^2 = w_{1,j} z_1 + w_{2,j} z_2 + \dots + w_{M,j} z_M$).

For most practical problems, evaluation of C requires solving high order integrals, which, if the dimensionality of the design space is sufficiently small, can be solved with techniques like tensor product Gauss-Legendre quadrature. Therefore, for complex engineering problems, like one studied in the present work, the estimation of C , \hat{C} , is evaluated using pseudo-random sampling techniques such as Monte Carlo or Latin hypercube sampling. Now the eigendecomposition of \hat{C} gives estimated eigenpairs, $\hat{\mathbf{W}}$ and $\hat{\Lambda}$:

$$C \approx \hat{C} = \hat{\mathbf{W}} \hat{\Lambda} \hat{\mathbf{W}}^T. \quad (7)$$

The accuracy of \hat{C} depends on the number of samples S , which is selected to have estimated eigenvalues $\hat{\Lambda}$ close to true eigenvalues Λ and it is measured as:

$$\omega = \text{dist}(\text{ran}(\mathbf{W}_1), \text{ran}(\hat{\mathbf{W}}_1)) = \|\mathbf{W}_1 \mathbf{W}_1^T - \hat{\mathbf{W}}_1 \hat{\mathbf{W}}_1^T\|. \quad (8)$$

C. Geometrically-Active Subspace

To create GAS, we assume data is distributed around a linear subspace of the original variables to learn the eigenvectors spanning this subspace. These eigenvectors are constructed from the eigendecomposition of the covariance matrix, C_g . For GAS, this matrix is constructed numerically by discretising \mathcal{G} into E elements of equal measure $\Delta\mathcal{G}$, which, for $n = 3$, gives spatial discretisation of $\delta(\zeta, \mathbf{x})$ as:

$$\mathbf{d}(\mathbf{x}) = \begin{bmatrix} \mathbf{d}_1(\mathbf{x}) \\ \mathbf{d}_2(\mathbf{x}) \\ \mathbf{d}_3(\mathbf{x}) \end{bmatrix} = \begin{bmatrix} d_{1,1}(\mathbf{x}) \\ \vdots \\ d_{1,E}(\mathbf{x}) \\ d_{2,1}(\mathbf{x}) \\ \vdots \\ d_{2,E}(\mathbf{x}) \\ \vdots \\ d_{3,1}(\mathbf{x}) \\ \vdots \\ d_{3,E}(\mathbf{x}) \end{bmatrix}, \quad (9)$$

where $\mathbf{d}(\mathbf{x})$ is a column matrix of size $[L \times 1]$ and $L = nE$. Using this discretisation, C^g can be obtained as:

$$C^g = \int_{\mathcal{X}} \mathbf{d}(\mathbf{x}) \mathbf{d}(\mathbf{x})^T \rho(\mathbf{x}) d\mathbf{x}. \quad (10)$$

As explained earlier, the analytical solution of C^g require solving high-dimensional integral, therefore, its approximation is obtained as in Eq. (11) while sampling \mathcal{X} with a statistically convergent number of Monte Carlo realisations, S_g , which creates dataset $\mathbf{X} = \{\mathbf{x}_r, r = 1, 2, \dots, S_g\} \sim \rho(\mathbf{x})$.

$$C^g = \hat{C}^g = \frac{1}{S_g} \sum_{r=1}^{S_g} \mathbf{d}(\mathbf{x}_r) \mathbf{d}(\mathbf{x}_r)^T = \hat{\mathbf{W}}^g \hat{\Lambda}^g (\hat{\mathbf{W}}^g)^T \approx \mathbf{W}^g \Lambda^g \mathbf{W}^{gT}. \quad (11)$$

Here, \mathbf{W}^g is $[L \times L]$ matrix whose columns are the eigenvectors. The eigenvalues Λ obtained from the eigendecomposition of C represents the variance resolved along the associate eigenvectors. The partition of \mathbf{W}^g into geometrically-active

(\mathbf{W}_1^g) and inactive (\mathbf{W}_2^g) feature matrices is done by looking for eigenvectors which retain minimum 95% of the geometric variance and this variance (σ^2) is measured as:

$$\sigma^2 = \frac{\sum_{i=1}^N \lambda_i}{\sum_{i=1}^M \lambda_i}, \quad (12)$$

and \mathbf{W}_1^g is a $[L \times N]$ matrix containing the first N components of \mathbf{W}^g . Its linear combination with $\mathbf{d}(\mathbf{x})$ creates the latent variable \mathbf{v} , which spans the basis of GAS, \mathcal{V} , of N dimensionality and is obtained as follows:

$$\mathbf{v} = \mathbf{W}_1^{gT} \mathbf{d}(\mathbf{x}) \in \mathcal{V} \subset \mathbb{R}^N. \quad (13)$$

The reduced order representation of $\mathbf{d}(\mathbf{x})$ is created with the active variable \mathbf{v} and represented as $\hat{\mathbf{d}}(\mathbf{v}) = \mathbf{W}_1^g \mathbf{v}$. Now the reconstruction accuracy in Eq. (2) can also be expressed in discrete form as:

$$MSE_g = \frac{1}{S_g} \sum_{r=1}^{S_g} \|\hat{\mathbf{d}}(\mathbf{v}_r) - \mathbf{d}(\mathbf{x}_r)\|^2, \quad (14)$$

where

$$\sigma_g^2 = \frac{1}{S_g} \sum_{r=1}^{S_g} \|\mathbf{d}(\mathbf{x}_r)\|^2. \quad (15)$$

Combining equation Eq. (14) and Eq. (15) gives NMSE as:

$$NMSE_g = \frac{\sum_{r=1}^{S_g} \|\hat{\mathbf{d}}(\mathbf{v}_r) - \mathbf{d}(\mathbf{x}_r)\|^2}{\sum_{r=1}^{S_g} \|\mathbf{d}(\mathbf{x}_r)\|^2}. \quad (16)$$

1. Physics-informed geometrically-active subspace

Based on [16], the geometric features can also be extracted containing variability of both geometry and physics. We refer a subspace formed with these features as *Physics-Informed Geometrically-Active Subspace* (PI-GAS). To construct this subspace, consider that along with the shape modification vector, $\delta \in \mathbb{R}^n$, there is a lumped (or global) physical parameter vector, $\varepsilon \in \mathbb{R}^{n^*}$ with $n^* = 1, \dots, \infty$, representing, e.g., drag, resistance, stability, etc. The domain of ε is defined with \mathcal{P} , which has a null measure and corresponds to an arbitrary point, ζ_θ , where the global physical parameter is virtually defined. Also, note that generally $\mathcal{Q} := \mathcal{G} \cup \mathcal{P}$ is not simply connected. Consider a combined vector $\gamma(\zeta, \mathbf{x}) \in \mathbb{R}^n$ with $n = \max\{n, n^*\}$ as:

$$\gamma(\zeta, \mathbf{x}) = \begin{cases} \delta(\zeta, \mathbf{x}) & \text{if } \zeta \in \mathcal{G} \\ \varepsilon(\zeta, \mathbf{x}) & \text{if } \zeta \in \mathcal{P} \end{cases} \quad (17)$$

Similar to GAS, here the aim of the dimensionality reduction is to identify a reduced-dimensionality representation $\hat{\gamma}(\zeta, \mathbf{v}^*)$. Here, \mathbf{W}_1^{g*} is a *physics-informed geometric feature* set of \mathcal{X} , which is obtained with eigendecomposition of covariance matrix C^{g*} given in Eq. (18). The linear combination of \mathbf{W}_1^{g*} with the discretisation of $\gamma(\zeta, \mathbf{x})$, $\mathbf{d}(\mathbf{x}^*)$, creates a *Physics-Informed Geometrically-Active Latent Variable* (PI-GALV) vector \mathbf{v}^* , which spans the PI-GAS (\mathcal{V}^*) of N^* dimensionality (see Eq. (19)).

$$C^{g*} = \int_{\mathcal{G}} \mathbf{d}(\mathbf{x}^*) \mathbf{d}(\mathbf{x}^*)^T \rho(\mathbf{x}) d\mathbf{x}, \quad (18)$$

$$\mathbf{v}^* = \mathbf{W}_1^{g*T} \mathbf{d}(\mathbf{x}^*) \in \mathcal{V}^* \subset \mathbb{R}^{N^*}. \quad (19)$$

D. Functionally-Active Subspace

Functionally-active subspace of \mathcal{V} or \mathcal{V}^* is developed with ASM strategy. In this section functional feature extraction of \mathcal{V} is presented and it follows the similar procedure for \mathcal{V}^* . Assume that QoI is a multivariate function, physics simulation or model f and $\nabla_{\mathbf{v}}f$ are the gradient of f with respect to \mathbf{v} . That is,

$$f = f(\mathbf{v}), \quad \nabla_{\mathbf{v}}f = \nabla_{\mathbf{v}}f(\mathbf{v}) = \begin{bmatrix} \frac{\partial f}{\partial v_1}(\mathbf{v}) \\ \vdots \\ \frac{\partial f}{\partial v_N}(\mathbf{v}) \end{bmatrix}, \quad \text{for all } \mathbf{v} \in \mathcal{V}. \quad (20)$$

The objective here is to find a P -dimensional functionally-active representation of geometrically-active subspace, where $P < N$. For this, the first step is to compute a covariance matrix C^f (given in Eq. (21)), which is defined as an average of the outer product of $\nabla_{\mathbf{v}}f$ with itself:

$$C^f = \int_{\mathcal{V}} (\nabla_{\mathbf{v}}f)(\nabla_{\mathbf{v}}f)^T \rho(\mathbf{v}) d\mathbf{v}. \quad (21)$$

Herein, f is assumed to be a square-integrable function with continuous partial derivatives with respect to \mathbf{v} . As mentioned before, the active directions are identified based on the eigendecomposition of the approximated covariance matrix in Eq. (21), which is obtained using following scheme:

$$C^f \approx \hat{C}^f = \frac{1}{S_f} \sum_{r=1}^{S_f} \nabla_{\mathbf{v}}f(\mathbf{v}_r) \nabla_{\mathbf{v}}f(\mathbf{v}_r)^T = \hat{\mathbf{W}}^f \hat{\Lambda}^f \hat{\mathbf{W}}^f{}^T \approx \mathbf{W}^f \Lambda^f \mathbf{W}^f{}^T, \quad (22)$$

where S_f is the number of designs points sampled from \mathcal{V} and the accuracy of \hat{C}^f depends on the distribution of S_f designs over \mathcal{V} .

\mathbf{W}^f is a $[N \times N]$ feature matrix, whose partition into the functionally-active (\mathbf{W}_1^f) and inactive (\mathbf{W}_2^f) features is done by looking for gaps in the elements of Λ . The separation of this kind is consistent with standard perturbation theory for eigenvector computations but is contrary to the heuristic used for the partition of geometrically-active and inactive features. For instance, if there is a larger gap between the λ_P and λ_{P+1} than between the λ_{P-1} and λ_P , then functionally-active estimation of \mathcal{V} is more accurate with first P elements of \mathbf{W}^f than its estimation with first $P-1$ elements of \mathbf{W}^f . This also implies that if $\Lambda_2^f \approx 0$ (i.e., $\{\lambda_{P+1}, \lambda_{P+2}, \dots, \lambda_M\} \approx 0$) then the mean-squared change in f along directions defined by the eigenvectors of \mathbf{W}_2^f is negligible. As f is continuous so the directional derivative $\nabla_{\mathbf{v}}f^T \mathbf{W}_2^f \approx 0$ everywhere in \mathcal{V} . In other words, f is constant/flat along the directions defined by the \mathbf{W}_2^f , which can be ignored for dimensionality reduction. Now, \mathbf{W}_1^f defines the functional feature set \mathbf{W}_1^f , whose linear combination with \mathbf{v} creates the FALV vector, \mathbf{u} , which spans the basis of the FAS, \mathcal{U} , as:

$$\mathbf{u} = \mathbf{W}_1^f{}^T \mathbf{v} = \mathbf{W}_1^f{}^T (\mathbf{W}_1^g \mathbf{d}(\mathbf{x})) \in \mathcal{U} \subset \mathbb{R}^P. \quad (23)$$

The reconstruction of the original GALV, $\hat{\mathbf{v}}$, can be obtained as

$$\hat{\mathbf{v}} = \mathbf{W}_1^f \mathbf{u} = \mathbf{W}_1^f \mathbf{W}_1^f{}^T \mathbf{v}, \quad (24)$$

where variable \mathbf{u} defines the reduced dimensionality representation $\hat{\delta}(\zeta, \mathbf{u})$ of shape modification vector $\hat{\delta}(\zeta, \mathbf{v})$ in functionally-active subspace, which is the reduced dimensional geometrically-active representation of shape modification vector $\delta(\zeta, \mathbf{x})$ in original design space \mathcal{X} . The reconstruction accuracy of this shape modification vector in functionally-active subspace is measured again with its discrete form, $\hat{\mathbf{d}}(\mathbf{u}) = \mathbf{W}_1^g \mathbf{W}_1^f \mathbf{u}$, as

$$NMSE_f = \frac{\sum_{r=1}^{S_f} \|\hat{\mathbf{d}}(\mathbf{u}_r) - \mathbf{d}(\mathbf{x}_r)\|^2}{\sum_{r=1}^{S_f} \|\mathbf{d}(\mathbf{x}_r)\|^2}. \quad (25)$$

Similarly, the FAS, \mathcal{U}^* , of \mathcal{V}^* is obtained after replacing \mathbf{v} with \mathbf{v}^* in Eq. (22). The basis of \mathcal{U}^* is defined by a latent variable \mathbf{u}^* , which again composed as linear combination of \mathbf{v}^* with physics-informed functional features $\mathbf{W}_1^{f^*}$.

E. Surrogate Modelling in Functionally-Active Subspace

The motivation for learning \mathcal{U} is to expedite the surrogate modelling for high-dimensionality design problems. These models suffer from the curse of dimensionality [17], therefore, reducing the input space dimensional, we can accept a small penalty in the accuracy of the f in exchange for the opportunity to tackle high-dimensionality. Therefore, after evaluating \mathbf{u} , f can be approximated in \mathcal{U} as

$$f(\mathbf{v}) \approx g(\mathbf{W}_1^f \mathbf{v}) = g(\mathbf{u}) \quad (26)$$

Afterwards, the surrogate model in \mathcal{U} can be built as:

$$g(\mathbf{u}) \approx g^*(\mathbf{u}) \equiv \mathcal{R}(\mathbf{u}; g_1, g_2, \dots, g_{S_r}). \quad (27)$$

where $g^*(\mathbf{u})$ is the surrogate model in \mathcal{U} and \mathcal{R} is the chosen response surface method trained on sample points g_1, g_2, \dots, g_{S_r} . The domain of g is

$$\mathcal{U} = \{\mathbf{u} = \mathbf{W}_1^f \mathbf{v}, \mathbf{v} \in \mathcal{V}\} \subset \mathbb{R}^P. \quad (28)$$

In this work, different surrogate modelling techniques have been used, including Gaussian Process Regression (GPR), for training and testing the surrogate models in the FAS. All these models have been widely utilised in the literature and will be discussed in Section IV.C. We have seen better results in term of training the surrogate model with GPR, thus, leaving the comprehensive details to [19, 20], below section gives brief deception on the key concepts and construction of GPR-based surrogate model.

1. Gaussian Process Regression

GPR is a non-parametric Bayesian approach [19], which have been used in different design applications [21]. It maps the nonlinear and globally coupled relationship between inputs and outputs sampled from a theoretically infinite-dimensional normal distribution and any finite number of samples in the input space, which follow a corresponding joint (multivariate) Gaussian distribution. The main advantages of GPR over other modelling techniques are, it can: (1) map relationship between inputs and outputs with small data size and, (2) easily handle noise in the data, thus, avoid over-fitting, and (3) optimise hyper-parameters from training data to increase the fit accuracy. For training, GPR assumes that the output y of g at input \mathbf{u} can be written as:

$$y = g(\mathbf{u}) + \epsilon, \text{ with } \epsilon \sim \mathcal{N}(0, \sigma_\epsilon^2). \quad (29)$$

Above is similar to Multiple Linear Regression (MLR) and assumes that an observation consists of an independent *signal* term $g(\mathbf{u})$ and a *noise* term ϵ . However, GPR assumes that $g(\mathbf{u})$ is a random variable and follows a particular distribution, which reflects our uncertainty regarding the function [20]. The uncertainty in $g(\mathbf{u})$ can be observed based on its output at different \mathbf{u} samples. The term ϵ represents the inherent randomness in the observations and is independent of the number of observations. In GPR, $g(\mathbf{u})$ assumes to be distributed as a Gaussian Process (GP):

$$g(\mathbf{u}) \sim GP(\mu(\mathbf{u}), k(\mathbf{u}, \mathbf{u}')). \quad (30)$$

A GP is any distribution over functions such that any finite set of function values $g(\mathbf{u}_1), g(\mathbf{u}_2), \dots, g(\mathbf{u}_{S_r})$ have a joint Gaussian distribution and is defined by a mean $\mu(\mathbf{u})$ and a covariance $k(\mathbf{u}, \mathbf{u}')$ function [20]. The $\mu(\mathbf{u})$ defines the expected function value at input \mathbf{u} ,

$$\mu(\mathbf{u}) = \mathbb{E}[g(\mathbf{u})], \quad (31)$$

Here, we set $\mu(\mathbf{u}) = 0$ to avoid any posterior computation, which is achieved by subtracting the mean from all observations. The term $k(\mathbf{u}, \mathbf{u}')$ in Eq. (30) defines dependence between the $g(\mathbf{u})$ and $g(\mathbf{u}')$ as:

$$k(\mathbf{u}, \mathbf{u}') = Cov[g(\mathbf{u}), g(\mathbf{u}')] = \mathbb{E}[\{g(\mathbf{u}) - \mu(\mathbf{u})\}\{g(\mathbf{u}') - \mu(\mathbf{u}')\}]. \quad (32)$$

Once $\mu(\mathbf{x})$ and $k(\mathbf{u}, \mathbf{u}')$ are chosen, we can use GP to draw priori and posterior function values upon previous observations. For that, we sample \mathcal{U} to obtain training dataset (\mathcal{D}^t) consisting of S_r samples;

$$\mathcal{D}^t = \{\mathbf{U}^t, \mathbf{g}^t\}, \quad (33)$$

where

$$\mathbf{U}^t = \begin{bmatrix} \mathbf{u}_1^t \\ \mathbf{u}_2^t \\ \vdots \\ \mathbf{u}_{S_r}^t \end{bmatrix} = \begin{bmatrix} u_{1,1}^t & u_{1,2}^t & \dots & u_{1,N}^t \\ u_{2,1}^t & u_{2,2}^t & \dots & u_{2,N}^t \\ \vdots & \vdots & \ddots & \vdots \\ u_{S_r,1}^t & u_{S_r,2}^t & \dots & u_{S_r,N}^t \end{bmatrix}, \quad \mathbf{g}^t = \begin{bmatrix} g(\mathbf{u}_1^t) \\ g(\mathbf{u}_2^t) \\ \vdots \\ g(\mathbf{u}_{S_r}^t) \end{bmatrix}. \quad (34)$$

Now, we want to make predictions for new inputs \mathbf{U}^* by sampling \mathbf{g}^* for the posterior distribution $\rho(g|\mathcal{D}_t)$. Moreover, by definition, \mathbf{g}^t and \mathbf{g}^* follows a joint multivariate normal distribution, which can be written as follows:

$$\begin{bmatrix} \mathbf{g}^t \\ \mathbf{g}' \end{bmatrix} \sim \mathcal{N} \left(0, \begin{bmatrix} K(\mathbf{U}^t, \mathbf{U}^t) + \sigma_\epsilon^2 \mathbf{I} & K(\mathbf{U}^t, \mathbf{U}') \\ K(\mathbf{U}', \mathbf{U}^t) & K(\mathbf{U}', \mathbf{U}') \end{bmatrix} \right). \quad (35)$$

Here, $K(\mathbf{U}^t, \mathbf{U}^t)$ is the covariance matrix,

$$K(\mathbf{U}^t, \mathbf{U}^t) = \begin{bmatrix} k(\mathbf{u}_1^t, \mathbf{u}_1^t) & k(\mathbf{u}_1^t, \mathbf{u}_2^t) & \dots & k(\mathbf{u}_1^t, \mathbf{u}_{S_r}^t) \\ k(\mathbf{u}_2^t, \mathbf{u}_1^t) & k(\mathbf{u}_2^t, \mathbf{u}_2^t) & \dots & k(\mathbf{u}_2^t, \mathbf{u}_{S_r}^t) \\ \vdots & \vdots & \ddots & \vdots \\ k(\mathbf{u}_{S_r}^t, \mathbf{u}_1^t) & k(\mathbf{u}_{S_r}^t, \mathbf{u}_2^t) & \dots & k(\mathbf{u}_{S_r}^t, \mathbf{u}_{S_r}^t) \end{bmatrix}, \quad (36)$$

between all observed points, and $K(\mathbf{U}', \mathbf{U}')$ is the covariance matrix between new points. $K(\mathbf{U}^t, \mathbf{U}')$ is the covariance matrix between the observed points and the new input points and $K(\mathbf{U}', \mathbf{U}^t)$ is vice versa. Where, in Eq. (35), parameters \mathbf{I} and σ_ϵ^2 are the identity matrix and noise level of observations, respectively. Following [20], the conditional distribution $\rho(\mathbf{g}'|\mathbf{U}^t, \mathbf{y}^t, \mathbf{U}')$ is then a multivariate normal distribution with mean (Eq. (37)) and covariance matrix (Eq. (38)).

$$\mu^t(\mathbf{u}) = K(\mathbf{u}, \mathbf{U}^t) [K(\mathbf{U}^t, \mathbf{U}^t) + \sigma_\epsilon^2 \mathbf{I}]^{-1} \mathbf{g}^t, \quad (37)$$

$$k_t(\mathbf{u}, \mathbf{u}') = k(\mathbf{u}, \mathbf{u}') - K(\mathbf{u}, \mathbf{U}^t) [K(\mathbf{U}^t, \mathbf{U}^t) + \sigma_\epsilon^2 \mathbf{I}]^{-1} K(\mathbf{U}^t, \mathbf{u}'). \quad (38)$$

In GPR, k is known as the *kernel* function [19] and a suitable choice of this function in based on assumptions such as smoothness and likely patterns to be expected in the training data [20]. For GPR, different types kernels have been developed and used, however, most common is the *exponential kernel function*,

$$k(\mathbf{u}, \mathbf{u}') = \sigma_s^2 \exp \left(\frac{-\|\mathbf{u} - \mathbf{u}'\|^2}{2\sigma_l^2} \right), \quad (39)$$

where σ_l^2 and σ_s^2 define the length-scale and signal variance, respectively, which can be altered to increase or reduce the priori correlation between points and the variability of the resulting function. $k(\mathbf{u}, \mathbf{u}')$ is often represented as $k(\mathbf{u}, \mathbf{u}'|\theta)$ to explicitly indicate its dependence on hyper-parameters, where θ is set containing all the hyper-parameters, σ_ϵ^2 , σ_s^2 and σ_l^2 (e.g., $\theta = (\sigma_\epsilon^2, \sigma_s^2, \sigma_l^2)$). These parameters are tuned for a specific dataset \mathcal{D} either with maximum likelihood approach [19] or Bayesian techniques [22], which measures how well a given θ describes \mathcal{D} . In this work, GPR with different kernel functions were tested and hyper-parameters were optimised with the Bayesian technique [22].

F. Optimisation

An optimisation process is carried out to explore \mathcal{U} for optimal design, during which designs are evaluated with the surrogate model. To initiate the optimisation we first need to define the bounding limits as $\mathbf{u}^l \leq \mathbf{u} \leq \mathbf{u}^u$, where \mathbf{u}^l and \mathbf{u}^u are vectors containing the lower and upper limits for \mathbf{u} and form a viable search space, which are evaluated based on heuristic presented in [17] and is given in Eq. (40).

$$\begin{aligned} \mathbf{u}^l &= \mathbf{x}^l \times \text{diag}(\text{sign}(\mathbf{W}_1^f)^T \mathbf{W}_1^f), \\ \mathbf{u}^u &= \mathbf{x}^u \times \text{diag}(\text{sign}(\mathbf{W}_1^f)^T \mathbf{W}_1^f), \end{aligned} \quad (40)$$

where $\text{sign}(\mathbf{W}_1^f)$ returns a matrix with sign of the components. Afterwards, to generate an optimal design, one can solve the following optimisation problem:

$$\begin{aligned}
& \text{given } \mathbf{u} \in \mathbb{R}^P \\
& \min_{\mathbf{u}} g(\mathbf{u}) \\
& \text{subject to } \mathbf{u} \in \mathcal{U} \\
& \quad \text{yield } \mathbf{u}_{optimal},
\end{aligned} \tag{41}$$

where, g is a surrogate model constructed from the initial S_r training samples, sampled from \mathcal{U} . The constraint $\mathbf{u} \in \mathcal{U}$ is a placeholder to ensure that $\mathbf{u}_{optimal}$ within the predefined boundary of the functionally-active subspace.

Further experimentation, visualisation or fabrication analyses of $\mathbf{u}_{optimal}$ might require evaluation of its full scale representation ($\mathbf{x}_{optimal}$) in the original design space X . Therefore, it is necessary to ensure that there lies a full scale repression of every $\mathbf{u} \in \mathcal{U}$, for which we solve a secondary optimisation problem at each evaluation of \mathbf{u} , which is defined as follows:

$$\begin{aligned}
& \text{given } \mathbf{u} = \mathbf{u}_{optimal} \\
& \hat{\mathbf{d}}(\mathbf{u}) = \mathbf{W}_1^g \mathbf{W}_1^f \mathbf{u}, \quad \mathbf{W}_1^g \in \mathbb{R}^{L \times N}, \quad \mathbf{W}_1^f \in \mathbb{R}^{N \times P} \\
& \mathbf{d}(\mathbf{x}_r) \text{ of } \delta(\zeta, \mathbf{x}_r) \text{ for } r^{th} \text{ design} \\
& \min_{\mathbf{x}} \tau_1 \times g(\mathbf{W}_1^{fT} (\mathbf{W}_1^{gT} \mathbf{d}(\mathbf{x}))) + \tau_2 \times \|\hat{\mathbf{d}}(\mathbf{u}) - \mathbf{d}(\mathbf{x})\|^2 \\
& \text{subject to } g(\mathbf{W}_1^{fT} (\mathbf{W}_1^{gT} \mathbf{d}(\mathbf{x}))) \leq g(\mathbf{u}) \\
& \quad \mathbf{x}^l \leq \mathbf{x} \leq \mathbf{x}^u \\
& \quad \text{yield } \mathbf{x}_{optimal}
\end{aligned} \tag{42}$$

The term $\|\cdot\|$ is an Euclidean norm, which facilitates exploration of $\mathbf{x}_{optimal}$ in close proximity of \mathbf{u}_{select} . Although $\mathbf{u}_{optimal}$ has already been found from optimisation in Eq. (41) but we have added $g(\cdot)$ in objective function of Eq. (42) in order to find if there is any further improvement that can be achieved during the exploration of X for $\mathbf{x}_{optimal}$. τ_1 and τ_2 adjust the weight of $g(\cdot)$ and $\|\cdot\|$ during optimisation and $\tau_1, \tau_2 \in [0, 1]$. As the main objective for this optimisation is to find $\mathbf{x}_{optimal}$, therefore, higher priority should be given to $\|\cdot\|$ by setting $\tau_2 > \tau_1$. The constraint $g(\mathbf{W}_1^{fT} (\mathbf{W}_1^{gT} \mathbf{d}(\mathbf{x}))) \leq g(\mathbf{u})$ in Eq. (42) ensures that $\mathbf{x}_{optimal}$ has either the same or higher performance than $\mathbf{u}_{optimal}$. Algorithm 1 gives the stepwise procedure for the implementation of the proposed approach.

III. Test Case

The proposed method is demonstrated for the hull-form optimisation of the DTMB 5415 model (see Fig. 3), an early and open to public version of the USS Arleigh Burke destroyer DDG 51, extensively used as an international benchmark for shape optimisation problems [23, 24]. Table 1 summarises the main characteristics of the hull and test conditions.

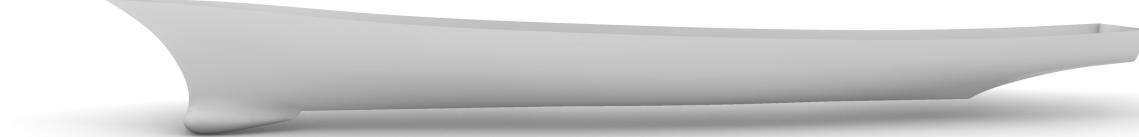


Fig. 3 CAD geometry of DTMB 5415 naval ship model used as test case for the proposed approach.

The optimisation aims at the reduction of the (model-scale) calm-water wave resistance coefficient (c_w) at Froude number equal to 0.25.

Algorithm 1 Step wise procedure of the proposed feature-to-feature learning approach.

- 1: Create an initial model \mathcal{G} composed of coordinate set $\zeta \in \mathcal{G} \subset \mathbb{R}^m$ and parameterise it with M design parameters (x_1, x_2, \dots, x_M) .
 - 2: Define the design space \mathcal{X} with lower and upper bounds of M parameters, $\mathcal{X} := \{x_k^l \leq x_k \leq x_k^u, \forall k \in \{1, 2, \dots, M\}\}$.
 - 3: Define a shape modification vector $\delta(\zeta, \mathbf{x})$ to modify \mathcal{G} for any realisation $\mathbf{x} \in \mathcal{X}$.
 - 4: Sample \mathcal{X} to create set $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{S_g}]^T$, where S_g is sample size.
 - 5: Evaluate shape modification set $\delta(\zeta, \mathbf{X}) = [\delta(\zeta, \mathbf{x}_1), \delta(\zeta, \mathbf{x}_2), \dots, \delta(\zeta, \mathbf{x}_{S_g})]^T$
 - 6: Create the dataset $\mathbf{D}(\mathbf{X})$ containing discrete representation of $\delta(\zeta, \mathbf{X})$ as $\mathbf{D}(\mathbf{X}) = [\mathbf{d}(\mathbf{x}_1), \mathbf{d}(\mathbf{x}_2), \dots, \mathbf{d}(\mathbf{x}_{S_g})]^T$, where the discretisation of i^{th} design \mathbf{x}_i is represented in Eq. (9).
 - 7: Compute C_g and its eigendecomposition: $C_g = \frac{1}{S_g} \sum_{r=1}^{S_g} \mathbf{d}(\mathbf{x}_r) \mathbf{d}(\mathbf{x}_r)^T = \mathbf{W}^g \Lambda^g \mathbf{W}^{gT}$
 - 8: Partition eigenspace of C_g as in Eq. (5), where elements of Λ_1^g capturing minimum 95% of the variance (i.e., $\sum_{i=1}^N \lambda_i / \sum_{i=1}^M \lambda_i \geq 95\%$).
 - 9: Form geometrically-active latent variable $\mathbf{v} = \mathbf{W}_1^g \mathbf{d}(\mathbf{x})$, $\mathbf{v} = \{v_r, r = 1, 2, \dots, N\}$.
 - 10: With \mathbf{v} create a geometrically-active subspace $\mathcal{V} \subset \mathbb{R}^N$ as, $\mathcal{V} := \{v_k^l \leq v_k \leq v_k^u, \forall k \in \{1, 2, \dots, N\}\}$, where v_k^l and v_k^u are the lower and upper bounds and $M < N$.
 - 11: Sample \mathcal{V} to create set $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_{S_f}]^T$, where S_f is sample size.
 - 12: Evaluate QoI for \mathbf{V} , $\mathbf{F} = [f(\mathbf{v}_1), f(\mathbf{v}_2), \dots, f(\mathbf{v}_N)]^T$
 - 13: Evaluate Gradients $\nabla \mathbf{F} = [\nabla_{\mathbf{v}} f(\mathbf{v}_1), \nabla_{\mathbf{v}} f(\mathbf{v}_2), \dots, \nabla_{\mathbf{v}} f(\mathbf{v}_{S_g})]^T$ with respect to \mathbf{v} .
 - 14: Compute C_f and its eigendecomposition: $\frac{1}{S_f} \sum_{r=0}^{S_f} \nabla_{\mathbf{v}} f(\mathbf{v}) \nabla_{\mathbf{v}} f(\mathbf{v}_q)^T = \mathbf{W}^f \Lambda^f \mathbf{W}^{fT}$
 - 15: Partition eigenspace of C_f as in Eq. (5), which is composed having maximum separation between last element of Λ_1^f and first element of Λ_2^f (i.e., $\lambda_P >> \lambda_{P+1}$).
 - 16: Form functionally-active subspace as $\mathbf{u} \in \mathcal{U} \subset \mathbb{R}^P$, which is bounded with lower \mathbf{u}^l and upper \mathbf{u}^u bounds (i.e., $\mathcal{U} := \{u_k^l \leq u_k \leq u_k^u, \forall k \in \{1, 2, \dots, P\}\}$).
 - 17: Sample \mathcal{U} to create set $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_{S_r}]^T$, where S_r sample size of the training dataset for building surrogate model.
 - 18: Evaluate QoI for \mathbf{U} , $\mathbf{G} = [g(\mathbf{u}_1), g(\mathbf{u}_2), \dots, g(\mathbf{u}_{S_r})]^T$
 - 19: Create dataset $\mathcal{D} = [\mathbf{U} \ \mathbf{G}]$ with columns of \mathbf{U} as independent variables and \mathbf{G} is the dependent variable.
 - 20: Use response surface method \mathcal{R} on dataset \mathcal{D} to create a surrogate model $g^*(\mathbf{u})$.
 - 21: Solve Eq. (41) to find an optimal design $\mathbf{u}_{optimal}$ in \mathcal{U} .
 - 22: To find the full space projection of $\mathbf{u}_{optimal}$ ($\mathbf{x}_{optimal}$) in \mathcal{X} solve optimisation problem in Eq. (42).
-

Table 1 DTMB 5415 original (model scale) hull main particulars.

Quantity	Symbol	Unit	Value
Displacement	∇	m^3	0.549
Length between perpendiculars	L_{pp}	m	5.720
Beam	B	m	0.760
Draft	T	m	0.248
Longitudinal center of gravity	LCG	m	2.884
Vertical center of gravity	VCG	m	0.056
Water density	ρ	kg/m^3	998.5
Kinematic viscosity	ν	m^2/s	1.09E-6
Gravity acceleration	g	m/s^2	9.803
Froude number	Fr	–	0.250

A. Shape Modification Method

The shape modification is defined using a recursive combination of $M = 27$ shape modification vectors over a hyper-rectangle embedding the demi hull:

$$\psi_i(\zeta) : \mathcal{A} = [0, L_{\zeta_1}] \times [0, L_{\zeta_2}] \times [0, L_{\zeta_3}] \in \mathbb{R}^3 \longrightarrow \mathbb{R}^3, \quad (43)$$

with $i = 1, \dots, M$. Specifically,

$$\delta(\zeta, \mathbf{x}) = \delta_M, \quad (44)$$

where

$$\delta_i(\zeta, \mathbf{x}) = x_i \psi_i(\zeta), \quad \text{with} \quad \begin{cases} \zeta = \zeta + \delta_{i-1} \\ \delta_1 = 0 \end{cases} \quad (45)$$

The coefficients $\{x_i, i = 1, \dots, M \in \mathbb{R}\}$ are the design parameters and forms a 27-dimensional initial (original) design space \mathcal{X} . For modification the shape functions are defined as:

$$\psi_i(\zeta) := \prod_{j=1}^3 \sin \left(\frac{a_{ij}\pi\zeta_j}{L_{\zeta_j}} + r_{ij} \right) \mathbf{e}_{q(i)}. \quad (46)$$

In Eq. (46), $\{a_{ij}, j = 1, 2, 3\} \in \mathbb{R}$ define the order of the function along j -th axis; $\{r_{ij}, j = 1, 2, 3\} \in \mathbb{R}$ are the corresponding spatial phases; $\{L_{\zeta_j}, j = 1, 2, 3\} \in \mathbb{R}$ are the hyper-rectangle edge lengths; $\mathbf{e}_{q(i)}$ is a unit vector. Modifications are applied along ζ_1 , ζ_2 , or ζ_3 , with $q(i) = 1, 2$, or 3 respectively. Details of setting parameters can be found in [16].

B. Hydrodynamic Solver and Setup

Hydrodynamic simulations are conducted using the code WARP (Wave Resistance Program), developed at CNR-INSEAN. Wave resistance computations are based on linear potential flow theory using Dawson (double-model) linearisation [25]. The frictional resistance is estimated using a flat-plate approximation, based on the local Reynolds number [26]. Details of equations, numerical implementations, and validation of the numerical solver are given in [27].

Simulations are performed for the right demi-hull, taking advantage of symmetry about the xz -plane. Fig. 4 shows the computational grid used for the simulation. The computational domain for the free-surface is defined within $1L_{pp}$ upstream, $3L_{pp}$ downstream, and $1.5L_{pp}$ sideways. A total of 75×20 grid nodes are used for the free surface, whereas 90×25 nodes are used for the hull discretisation.

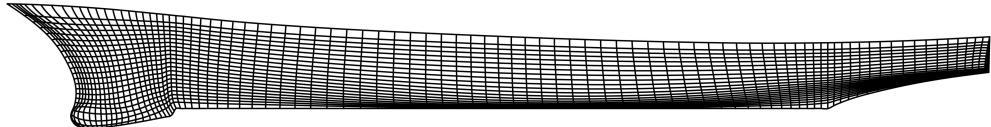


Fig. 4 Computational grid used for numerical implementation of feature extraction and for the potential flow simulation.

IV. Results and Discussion

To validate feature-to-feature learning we tested three different SDO pipelines. The sequential workflow of these pipelines is shown in Fig. 5. The first pipeline is the conventional ASM, which learns the lower dimensional FAS, \mathcal{U}_1 , from gradients of c_w ($\Delta_x c_w$) evaluated with respect to design parameters. The second pipeline is based on the proposed two-step feature-to-feature learning. In this pipeline, first, a GAS representation of the original design space is learned. Then, functional features of this subspace are extracted with eigendecomposition of the covariance matrix composed of gradients ($\Delta_v c_w$), which are learned as a function of geometric features. This decomposition creates a new FAS, \mathcal{U}_2 . Finally, in the third pipeline, the PI-GAS is created first, which contains the variability of both geometry and c_w . Afterwards, functionally-active representation (\mathcal{U}_3) of this subspace is extracted to further reduced its dimensionality.

Once \mathcal{U}_1 , \mathcal{U}_2 and \mathcal{U}_3 are created then, respectively, we build three surrogate models g_1 , g_2 and g_3 , which are connected with optimiser to find an optimal design for the test case. The efficiency of these pipelines is evaluated in

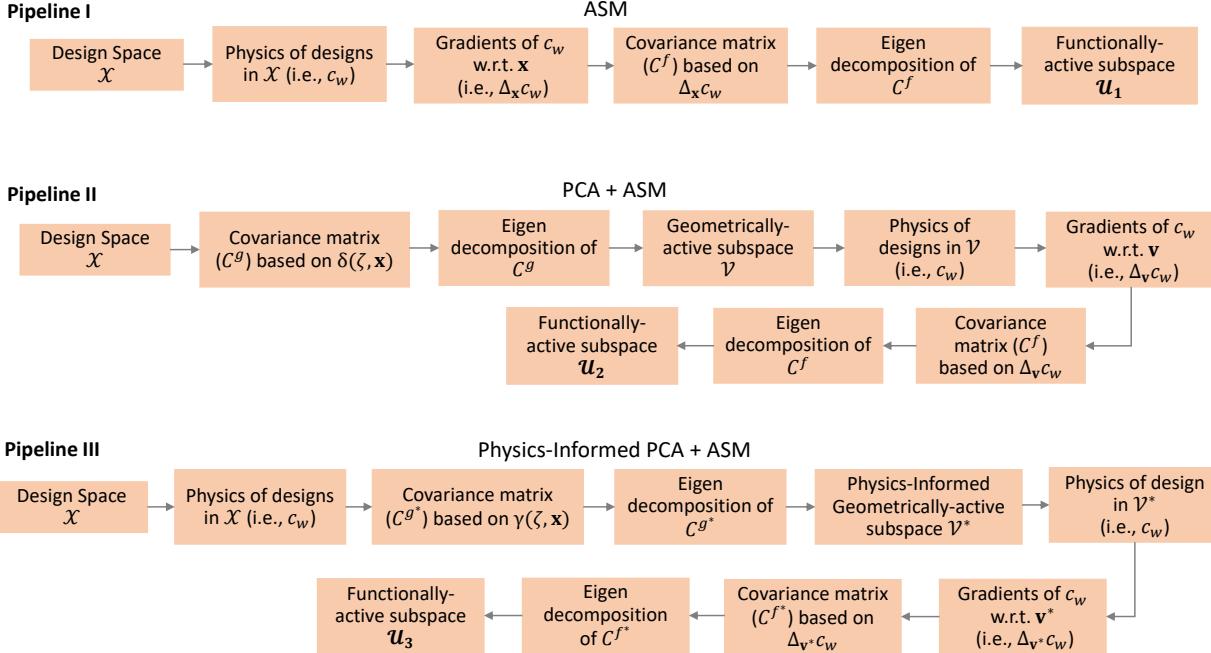


Fig. 5 Sequential work flow of testing pipelines used for the experimentation and validation of the proposed approach.

term of final dimensionality of the respective subspace, the accuracy of the surrogate model, and the ability of the optimiser to find an optimal solution with the least computational cost.

In the subsequent sections, we first provide numerical results on the extraction of the geometric and physics-informed geometric features of the original design space, which creates latent variables for the construction of GAS and PI-GAS. Afterwards, followed by the discussion on surrogate modelling and shape optimisation, we explain the results of the FAS formed with original design parameters and compare it with those formulated with two-step feature extraction.

A. Dimensionality reduction with geometric features

Extraction of geometric features to form \mathcal{V} or \mathcal{V}^* is commenced with the eigendecomposition of C^g or C^{g*} , which accumulates the geometric variability of designs into eigenvalues and eigenvectors. The former identifies the original geometric variability of the full design space in term of geometry or physics, whereas later contains the extracted feature for the latent variable and defines the shape deformation modes of highest impact on the hull geometry. Here, the number, N and N^* , of latent directions for \mathcal{V} and \mathcal{V}^* subspaces are set to achieve the $NMSE \leq 5\%$, resulting in active latent variables capturing 95% variability of the original design space.

As mentioned before, the eigendecomposition is implemented on a discrete version of the covariance matrix, for which the original design space (\mathcal{X}) was sampled following a uniform random distribution of $S_g = S_f = S_r = 9000$ hull-form designs by the Monte Carlo method, according to parametric studies performed in earlier work [15]. Afterwards, the elemental discretisation of the sampled designs was generated with each design containing $L = n[90 \times 25]$ elements. This created a training dataset for feature extraction used for the construction of \mathcal{V} . To form \mathcal{V}^* , c_w values of designs were also included in this dataset, which was evaluated under the settings mentioned in section III.B. Figure 6 (a) and (b), show the absolute and percentage of the geometric variance retained, which is evaluated as a cumulative sum of their associated eigenvalues for \mathcal{V} and \mathcal{V}^* , respectively.

From Figure 6 it can be seen that for the present test case under the specified setting that the first 15 and 11 eigenvectors, $\mathbf{W}_1^g = \{\mathbf{w}_i^g, i = 1, 2, \dots, 15\}$ and $\mathbf{W}_1^{g*} = \{\mathbf{w}_i^{g*}, i = 1, 2, \dots, 15\}$, obtained from the eigendecomposition of C^g and C^{g*} , retains 95.18% and 95.34% of geometric variance, respectively. This results in approximately 44% ($N = 15$) and 59% ($N^* = 11$) reduction of original design space's dimensionality to create 15- and 11-dimensional \mathcal{V} and \mathcal{V}^* , respectively. This also quantifies that there are only $N = 15$ or $N^* = 11$ latent directions that govern or accumulates all the geometric variations in \mathcal{X} . It is also noteworthy that in the presence of c_w the higher geometric variance is achieved

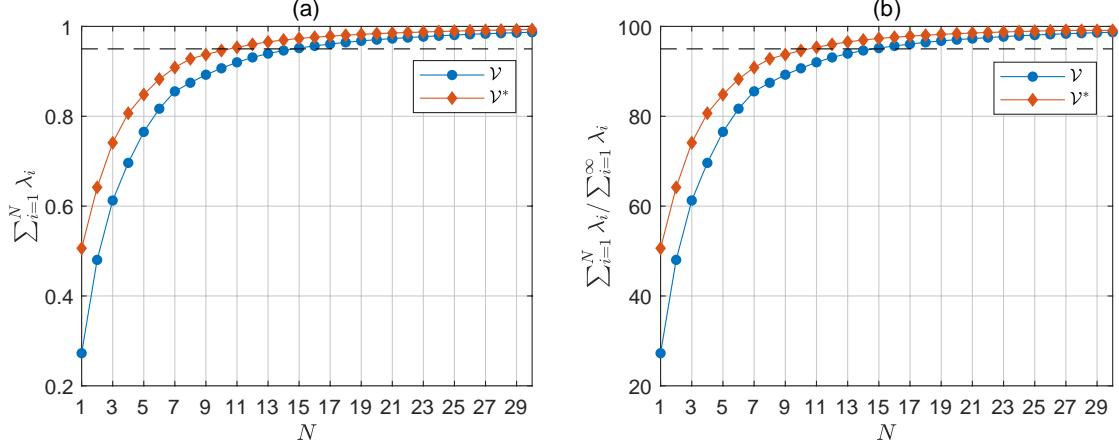


Fig. 6 Plot showing (a) absolute and (b) percentage of the geometric variance retained by geometrically-active (\mathcal{V}) and physics-informed geometrically active (\mathcal{V}^*) subspaces of dimension N . Horizontal lines on plot (a) and (b) indicates the threshold for 95% of the geometric variance.

and its first eigenvector captures 50.61% of the variance, which is approximately twice than the one captured by the first eigenvector of C^g . Moreover, up to $N, N^* = 20$, eigenvectors obtained from C^{g^*} contains higher cumulative sum of eigenvalues compare to ones obtained from C^g , however, it tends to be the same after 20th eigenvector.

Figures 7 (a) and (b) shows first three eigenvectors, $\{\mathbf{w}_i^g\}_{i=1}^3$ and $\{\mathbf{w}_i^{g^*}\}_{i=1}^3$, on the hull surface obtained for \mathcal{V} and \mathcal{V}^* , respectively. The visualisation of these eigenmodes directly on the design surface delivers an insight of great practical value as it shows the type and order of the variance accounted for the most important eigenvector on the ship hull geometry during shape modification. Both sets of eigenvector exhibit, to some extent, a global shape modification on the geometry, while accumulating significant variations in sectional area and waterline along the ship length.

To compare the eigenmodes of \mathcal{V} and \mathcal{V}^* , a statistical metric, commonly known as Modal Assurance Criterion (MAC) [28], is used. MAC is a widely used metric to measure the statistical consistency or similarity between eigenmodes obtained analytically and experimentally. It is bounded between 0 and 1, with 1 indicating two mode shapes are similar. The results of MAC evaluated with the first three eigenvectors of \mathcal{V} and \mathcal{V}^* are shown in Fig. 8. It is interesting to note that in the physics-informed formulation, the first eigenvector ($\mathbf{w}_1^{g^*}$) has been changed due to the presence of c_w , however, its second ($\mathbf{w}_2^{g^*}$) and third ($\mathbf{w}_3^{g^*}$) eigenvector shares high degree of similarity with \mathbf{w}_1^g and \mathbf{w}_2^g . In addition, $\mathbf{w}_1^{g^*}$ and $\mathbf{w}_3^{g^*}$ also show some similarity. These results provide interesting insight and again quantifies that the eigenvectors for \mathcal{V}^* captures more the geometric variance of \mathcal{X} . This is resulted because of the c_w , which may act as the shape descriptor for the sampled designs providing additional information to capture a higher percentage of geometric variance with fewer latent variables. However, compared to \mathcal{V} , \mathcal{V}^* is computationally expensive to evaluate as it requires the evaluation of c_w . One could study how geometrically-active features will behave if one could use the geometry-based descriptor, such as shape integrals, which are computationally inexpensive to evaluate compare to physics, into the dataset along with the designs' parametric discretisation. Although this would require the validation of integrals' accuracy and decision on satiable order of integral to achieve a concrete descriptor, nevertheless, this seems to be an interesting study, and authors plan to work in the future.

B. Dimensionality reduction with functional features

As explained previously, the extraction of functional features of given design space is driven with the eigendecomposition of the covariance matrix composed of the gradients of c_w with respect to the design parameter or the latent variables, if the functional feature of \mathcal{V} or \mathcal{V}^* have to be learnt.

1. Learning Gradients

To evaluate gradients of QoI, different techniques have been proposed in the literature, for instance, if numerical noise in the QoI is small enough then one could use finite-difference or could evaluate gradients using adjoint solvers if

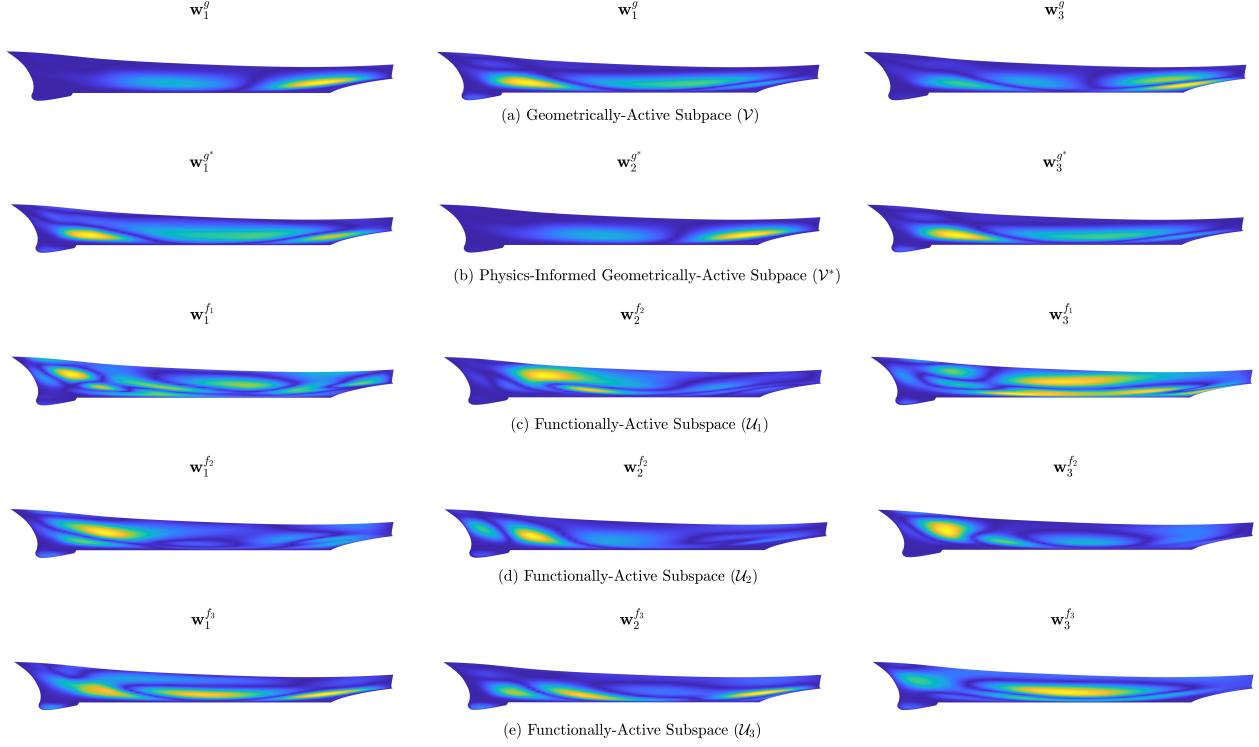


Fig. 7 Shape deformation modes corresponding to (a) first, (b) second and (c) third eigenvectors obtained for geometric and functional variance retained by geometrically-active (\mathcal{V}), physics-informed geometrically active (\mathcal{V}^*), and functionally-active subspaces; \mathcal{U}_1 , \mathcal{U}_2 and \mathcal{U}_3 . The colours indicate the magnitude of surface normal displacement.

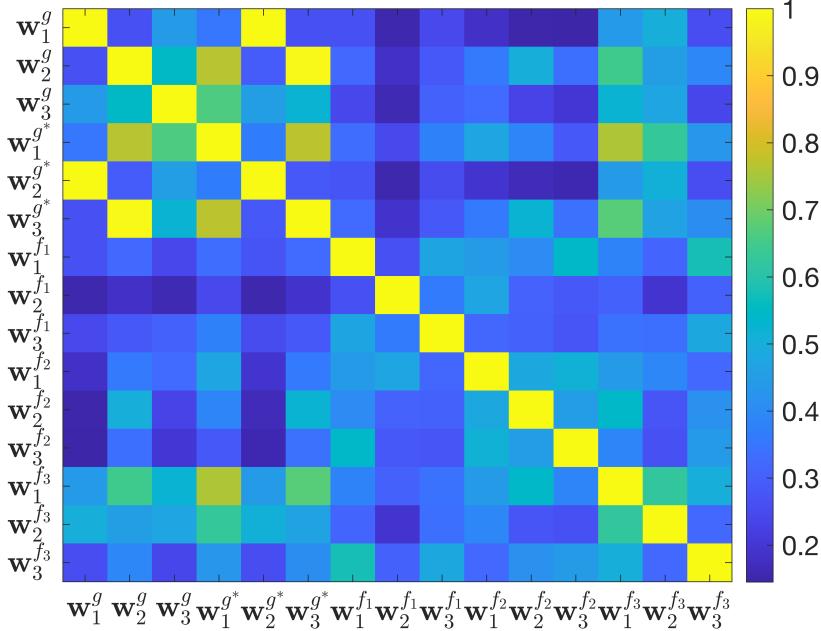


Fig. 8 Modal assurance criterion plot to compare the first three eigenvectors of the geometrically-active, physics-informed geometrically-active and functionally-active subspaces.

the base line legacy code or simulation allows. However, often these capabilities are absent when simulation includes multiple or coupled components, like in the present case. Moreover, for a complex problem involving freeform shapes like cars and ships, the gradients, especially evaluated from adjoint solvers, tends to be qualitatively correct, they have right relative scaling and right sign. However, quantitatively, in term of magnitude, they are not reliable [29, 30], at least when they are used for active subspace.

Therefore, one could approximate these gradients with a local surrogate model, as we are interested, for active subspace, in the local behaviour of the problem when approximating gradients. Therefore, inventors of ASM proposed an algorithm of building *local model-based* gradients, in which a local linear model is fitted with a subset of predictions from the training dataset [18].

In this approach, to evaluate the gradients for an i^{th} design $\mathbf{x}_i/\mathbf{v}_i/\mathbf{v}_i^*$ we obtain a subset of ℓ designs from training dataset nearest to $\mathbf{x}_i/\mathbf{v}_i/\mathbf{v}_i^*$ along with their c_w values. A surrogate model is fitted on this subset and the gradients of this model are evaluated. This process is repeated with all the samples in the training dataset. [18] recommends to use least squares to fit a local MLR model, but from our experiments, we found a local GPR model with linear basis fits the subset with better accuracy and thus utilised in the present study. Figure 9 (a) and (b), show the sufficient summary plot for the MLR and GPR models deployed on a subset containing $\ell = 700$ samples from \mathcal{V} and gradients evaluated form these models for the baseline hull design are shown in Figure 9 (c). Setting an appropriate value for ℓ also require some attention, which we will discuss in the next paragraph. The R^2 and MSE for GPR model are 0.97152 and 0.16709, and for MLR model these values are 0.66291 and 0.58155, respectively. From Figure 9 (a) and (b) it is clear that GPR well approximates the problem, therefore, gradients evaluated from this model are of better quality. Analysing Figure 9 (c), it can be seen for most of the parameters, gradients evaluated from MLR are lower in magnitude compare to the ones evaluated from GPR. Moreover, for the parameters 6, 13 and 14, gradients form MLR have also opposite signs compare to gradients evaluated from GPR. Therefore, for functional features in all three test cases, we used model-based gradients approximated with local GPR.

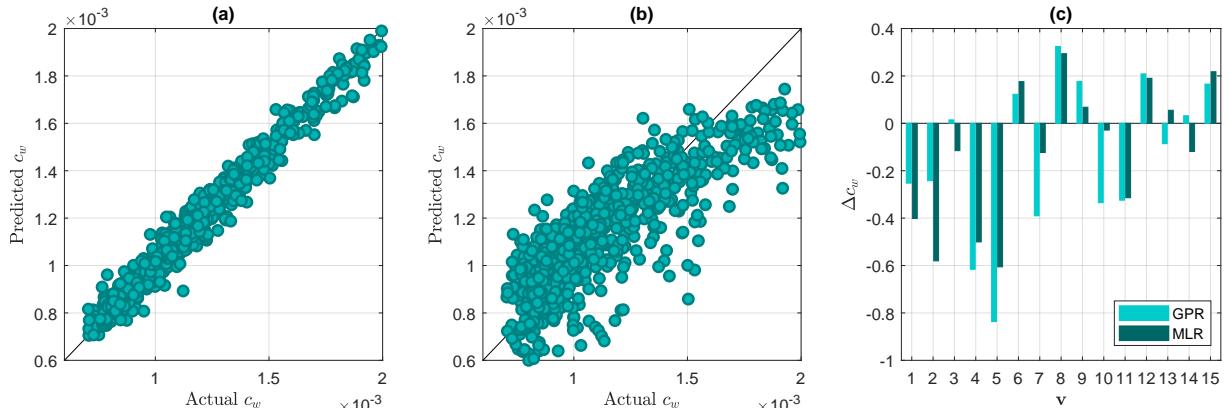


Fig. 9 Sufficient summary plot between actual and predicted wave resistance coefficients (c_w) evaluated using Gaussian Process Regression (GPR) and Multiple Linear Regression (MLR). (b) Comparison of model-based gradients obtained using GPR and MLR models.

As mentioned in the previous paragraph the value of ℓ has also some impact on the model accuracy thereby on its gradients, as it defines the number of samples required to be in the subset for the construction of a reliable local model. Based on the experiments, [18] proposed that the value for ℓ should be chosen greater than the number of parameters defining the model and less or equal to the total number of samples in the training dataset (i.e., $n < \ell \leq S$). However, from the experiments performed in this work an instability was observed in the model-based gradients as the value for ℓ varied, especially when the value of ℓ is close to n . Therefore, to choose an appropriate value we run an experiment where ℓ was varied iteratively and at each iteration, a model was developed and its gradients were evaluated. This process was repeated and stopped when gradients became stable. The results of this experiment performed during the construction of \mathcal{U}_2 can be seen in Figure 10. From this Figure it can be observe that from $\ell = 50$ to $\ell = 200$ gradients of c_w , Δc_w , fluctuates greatly, after $\ell = 200$ they start to converge till $\ell = 700$ and from this point, gradients tend to be stable. Therefore, we chooses ℓ equals to 750, 700 and 400 for \mathcal{U}_1 , \mathcal{U}_2 and \mathcal{U}_3 , respectively. Overall, after a certain ℓ values, the model-based gradients were quite constant throughout the sampled space or subspace for three test pipelines.

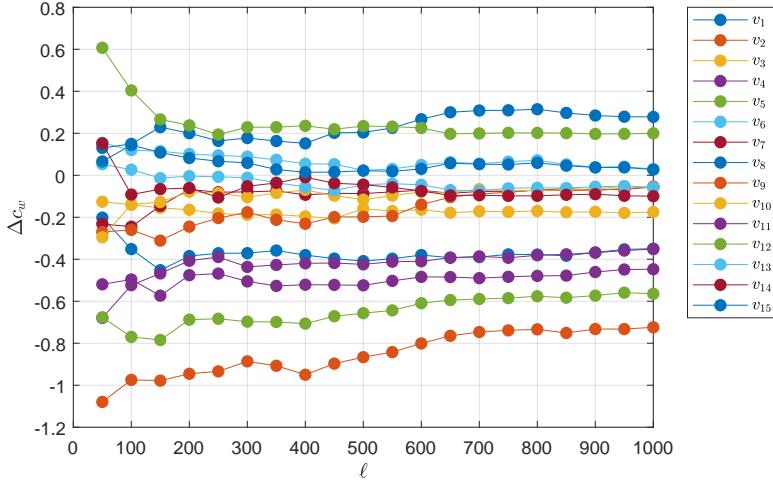


Fig. 10 Plot for gradients of wave resistance coefficient (Δc_w) versus the size of training dataset (ℓ) used to construct the GPR model for the evaluation of model-based gradients of the baseline test model.

2. Extracting Features

As explained earlier for FAS, the decision on the separation between active and inactive features is made by inspecting the decay in eigenvalues and taking the first P feature vectors having the highest separating between their eigenvalues. These P features create latent variables for a P -dimensional FAS.

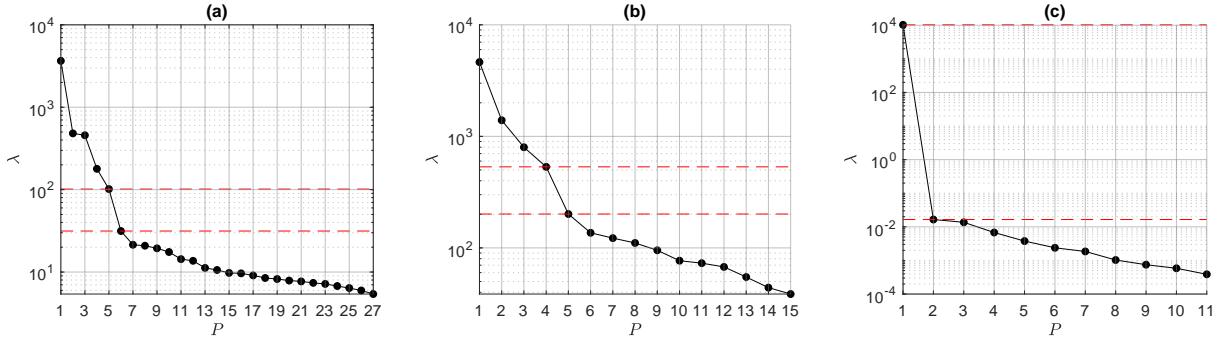


Fig. 11 Eigenvalue decay versus the dimensionality (P) of functionally-active subspace representation of (a) original design space, (b) geometrically-active and (c) physics-informed geometrically-active subspaces.

Fig. 11 (a), (b) and (c) shows the plot of eigenvalues obtained during the feature extraction for \mathcal{U}_1 , \mathcal{U}_2 and \mathcal{U}_3 , respectively, in three pipelines illustrated in Figure 5. The plot in Fig. 11 (a) shows the prominent separation between first and second eigenvalues and between fifth and sixth eigenvalues, respectively, this shows the potential for a one- or five-dimensional \mathcal{U}_1 . The reconstruction error is high if \mathcal{U}_1 is taken as one-dimensional, therefore, based on second prominent separation, first five eigenvectors $\mathbf{W}_1^{f_1} = \{\mathbf{w}_i^{f_1}, i = 1, 2, \dots, 5\}$ are used to create a five-dimensional (i.e., $P_1 = 5$) \mathcal{U}_1 . This results in approximately 81% reduction of the original design space's dimensionality.

Fig. 11 (b) shows the eigenvalue plot for the second test pipeline, which is the two-step feature-to-feature learning. In this case, functional features of \mathcal{V} are extracted, which further reduces its dimensionality. From Fig. 11 (b), it can be seen that the largest separation occurs between forth and five eigenvalues. This results in a $P_2 = 4$ -dimensional \mathcal{U}_2 , span by the first four eigenvectors $\mathbf{W}_1^{f_2} = \{\mathbf{w}_i^{f_2}, i = 1, 2, \dots, 4\}$, resulting in 85% reduction in the dimensionality of \mathcal{X} . It can be seen that, in this case, compare to \mathcal{U}_1 , there is no notable reduction is achieved with two-step feature extraction, however, as we will see in subsequent sections, that in term of surrogate modelling and optimisation, \mathcal{U}_2 captures better geometric and functional variability.

As explained previously, in the third test pipeline, first geometric decomposition happens not only in term of

geometry but also in term of c_w , which creates \mathcal{V}^* . Afterwards, functional features of this subspace are extracted and the eigenvalues obtained during this extraction are shown in Fig. 11 (c). These results show a significant separation between first and second eigenvalue, resulting in one-dimensional \mathcal{U}_3 (i.e., $P_3 = 1$) with first eigenvector $\mathbf{W}_1^{f_3} = \{\mathbf{w}_1^{f_3}\}$. The significant separation between first and second eigenvalue happens because the basis of the physics-informed geometric representation of original space accounts for the variation of both design parameters and c_w . As the first principal direction (eigenvector with the highest eigenvalue, i.e., $\mathbf{w}_1^{g^*}$) account for the maximum variance in this subspace, therefore, when this is projected on the functionally-active latent variables (\mathbf{u}^*), which are learned with the gradients of c_w with respect to \mathbf{v}^* , so the most of the variance in c_w is captured with $\mathbf{w}_1^{f_3}$.

Figure 7 (c), (d) and (e) show the eigenmodes of \mathcal{U}_1 , \mathcal{U}_2 and \mathcal{U}_3 , respectively, and the comparison between these eigenmodes is shown in term of MAC plot in Figure 8. First insight we try to make from these results is to look if eigenvectors of $\mathbf{W}_1^{f_1}$, $\mathbf{W}_1^{f_2}$ or $\mathbf{W}_1^{f_3}$ share some similarities with \mathbf{W}_1^g or $\mathbf{W}_1^{g^*}$. It can be observed that first three eigenvectors of $\mathbf{W}_1^{f_1}$ and $\mathbf{W}_1^{f_2}$ does not share any similarity as their MAC values is below 0.5, however, eigenvectors of $\mathbf{W}_1^{f_3}$, especially $\mathbf{w}_1^{f_3}$, show some resemblance. It is also interesting to note that $\mathbf{w}_1^{f_3}$ are $\mathbf{w}_1^{g^*}$ are alike and is also slightly similar to $\mathbf{w}_3^{g^*}$ and \mathbf{w}_2^g . Moreover, it is also noteworthy that none of the elements of $\mathbf{W}_1^{f_1}$, $\mathbf{W}_1^{f_2}$ and $\mathbf{W}_1^{f_3}$ does not have significant likeness.

3. Sensitivity analysis

The functional features can give useful insight in term of activity/sensitivity scores interpretation of original design parameters or latent variables on c_w . These sensitivity scores are similar to that one would obtain from Sobol' total sensitivity indices and derivative-based global sensitivity measures [31]. The eigenvectors identify the most important direction X , which implies that QoI, c_w in our case, changes the most along the latent variables defined by these eigenvectors. The components of each eigenvector measure the relative change along the direction defined by this eigenvector, so they impart significance to each of the design parameters of the original design space or latent variables of GAS or PI-GAS. Therefore, a global sensitivity analysis metric can be defined as:

$$\phi_i = \phi_i(P) = \sum_{j=1}^P \lambda_j w_{i,j}, \text{ where, } i = 1, 2, \dots, P \quad (47)$$

Here, ϕ_i is the activity score of the i^{th} parameter indicating relative significance on c_w . For better visualisation, we normalise the activity scores using:

$$\phi_i \mapsto \frac{\phi_i}{\sqrt{\sum_{i=1}^P \phi_i^2}}. \quad (48)$$

For further details, interested readers should refer to ref. [31]. Figure 12 (a), (b) and (c) shows the activity score for \mathbf{x} , \mathbf{v} and \mathbf{v}^* , respectively. From Figure 12 (a), it can be observe that the most significant parameter for X is x_{14} followed by x_8 , x_{15} and x_4 , where x_1 is the least significant parameter. In case of \mathcal{V} (Figure 12 (b)), the latent variable v_2 is the most significant one and v_{15} is the least significant parameter. Moreover, it is interesting to note that in the physics-informed formulation (Figure 12 (c)), the first latent variable v_1^* is the most significant one and its magnitude is approximately more than 90% higher than the rest of the latent variables. This again shows an interesting behaviour of this design space, when physics is involved during geometric feature extraction, then the first eigenvector covers all the functional variance and v_1^* of this eigenvector is responsible for the maximum change in c_w . This indicates that in this case, the first eigenvector has dominated or over-learned all the features in terms of physics.

C. Surrogate Model Training

The training data for surrogate models consists of $S_r = 9000$ uniformly distributed designs with design parameters or latent variables as independent variables and c_w as the dependent variable. The accuracy of the surrogate model can be effected by the outliers in the training dataset, which can result inaccurate or mediocre prediction of c_w . Therefore, a univariate method based on the quartiles is used to detect and remove outliers [32], which identifies an element of the dataset as an outlier if its value is more than 1.5 interquartile ranges above the upper quartile (75%) or below the lower quartile (25%). GPR was then applied to build surrogate models g_1 , g_2 and g_3 , with samples from \mathcal{U}_1 , \mathcal{U}_2 and \mathcal{U}_3 , respectively, and hyper-parameters of these models are optimised with different kernel function but square exponential kernel provided better results thus selected for the final surrogate model. The comparative results of training GPR with

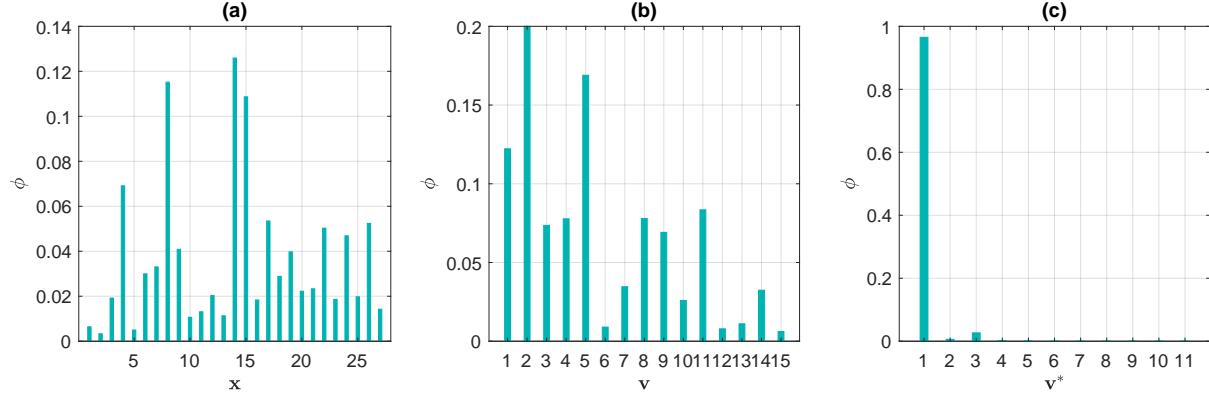


Fig. 12 Plot showing activity/sensitivity scores (ϕ) for the (a) design parameters, x , (b) geometrically-active latent variables, v and (c) physics-informed geometrically-active latent variables, v^* .

Table 2 Error and hyper-parameter values obtained during the training of surrogate models g_1 , g_2 and g_3 in functionally-active subspaces \mathcal{U}_1 , \mathcal{U}_2 and \mathcal{U}_3 .

Surrogate models	g_1	g_2	g_3
Dimensions	5	4	1
Training-MSE	0.31594	0.20619	1.7499e-05
Cross-validation-MSE	0.33141	0.21993	1.9108e-05
R^2	0.68402	0.80379	0.99998
Hyper-parameters			
σ_ϵ^2	0.07330	0.13790	0.14440
σ_l^2	0.00080	0.64871	3.45700
σ_s^2	1.48940	9.83550	5.63820

different kernel function will be discussed in the subsequent sections. For training validation, 10-fold cross-validation is implemented. Table 2 shows the results of training-MSE, which is evaluated between actual and predicted values of c_w in the training dataset, and cross-validation-MSE is calculated when unseen data is given to the model for prediction. The optimised values of the hyper-parameters for the surrogate models are also given in Table 2,

In these cases, cross-validation- and training-MSE values are very close to each other which ensures a good generalisation capability of the trained models. The R^2 error is bounded between zero and one and its value equal to zero indicates that the trained model does not improve prediction over the mean model, and value close to one shows good prediction, thus a model with higher R^2 should be selected. From the results of Table 2 one can see that g_3 , which is constructed from \mathcal{U}_3 , has the lowest training error followed by g_2 and g_1 . These results show that feature-to-feature dimensionality reduction does help to elevate the surrogate modelling accuracy. Figure 13 (a), shows the plot of MSE versus dimensionality of g_1 , g_2 and g_3 . It is note worthy that for all three models the MSE decreases as their dimensionality increase, however, for g_3 this change is negligible (see Figure 13 (b)) as even with one parameter g_3 has significantly higher accuracy compare to other two models.

In the case of g_1 , the MSE decreases up to five parameters and after that there is no significant improvement. Note that these results also aligns with results of eigendecomposition that the whole problem can be well represented with five latent variables. Similar behaviour can be observed in the case of g_2 , the MSE continues to drop up to four parameters and there is no significant improvement with further increasing its dimensionality. One might be interested in this slight improvement in the model with increasing the dimensionality, however, this will increase the computational cost for training the model. For instance, on a basic personal computer with an i7-7700 Intel Core, 3.6-GHz processor, and 8-GB physical memory, it took 20.68 and 55.16 minutes to train a five- and 27-dimensional g_1 , respectively.

It should also be noted that the MSE for g_2 remains lower compared to g_1 , even if it is constructed with all parameters. Moreover, as explained previously, even though there is no significant difference between \mathcal{U}_1 and \mathcal{U}_2 in term of

dimensionality reduction but g_2 with four latent variables has notably higher accuracy than g_1 trained with five latent variables obtained from directly from \mathcal{X} (i.e., single-step learning). This again demonstrates the potential of two-step feature-to-feature learning for surrogate modelling.

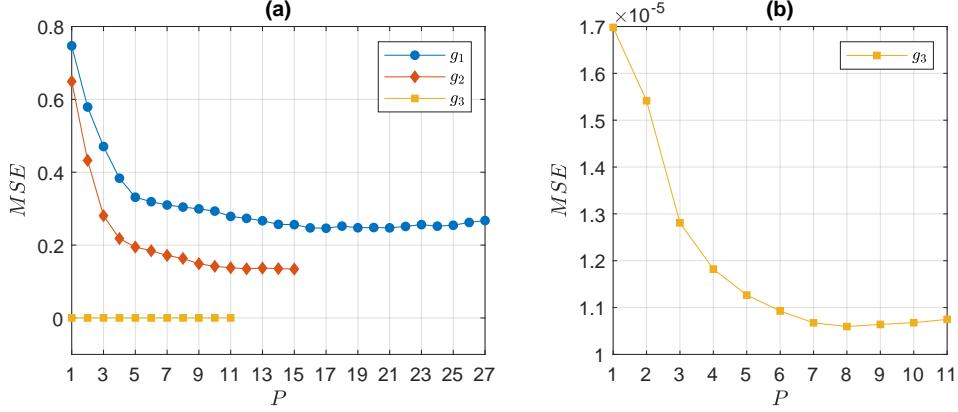


Fig. 13 Plot showing Mean Square Error (MSE) versus dimensionality (P) of surrogate models g_1 , g_2 and g_3 trained, respectively, with designs sampled from \mathcal{U}_1 , \mathcal{U}_2 and \mathcal{U}_3 . (b) Magnified version of plot between MSE and dimensionality of g_3 .

1. Comparison of GPR-based surrogate model with other techniques

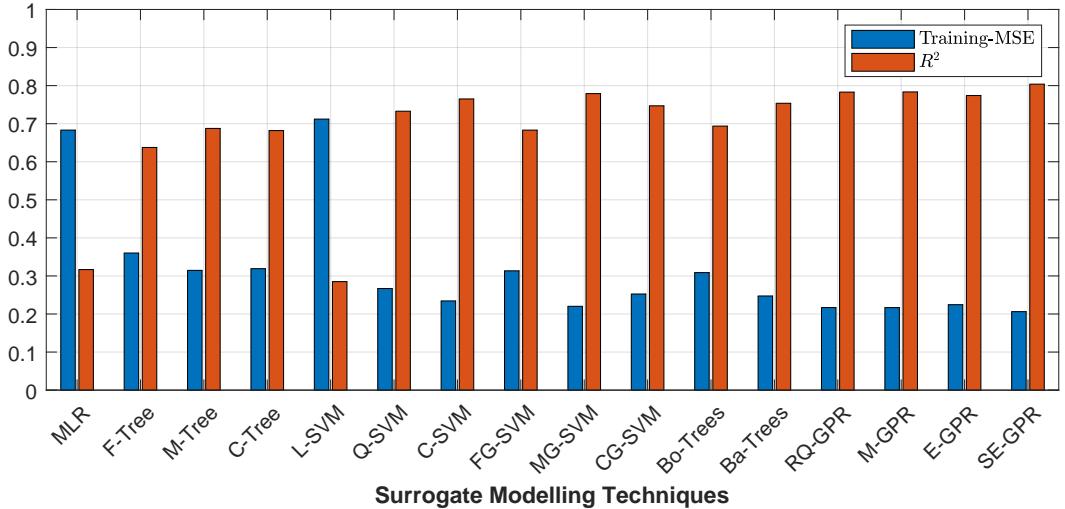
The performance of the GPR-based surrogate model was also compared with other meta-modelling techniques, such as MLR, Decision Regression Trees (DRT), Support Vector Machine (SVM), along with their different variations. In DTR, we tested fine, medium and coarse versions, in which leaf size was set to 4, 12 and 36 and SVM was tested with different kernels, including, linear, quadratic, cubic and Gaussian kernels with varying length-scale (σ_l^2) resulting in fine ($\sigma_l^2 = \sqrt{S_r}/4$), medium ($\sigma_l^2 = \sqrt{S_r}$) and coarse ($\sigma_l^2 = 4\sqrt{S_r}$) Gaussian SVM. Moreover, the performance of the GPR model trained with different kernel functions, rational quadratic, matern 5/2, exponential and squared exponential, was also evaluated in-term of training-MSE and R^2 . The results of this comparison for g_2 are given in Fig 14, which show an interesting behaviour that the linear techniques; MLR and Linear SVM, cannot maps well the nonlinear behaviour of the problem thus compare to other techniques they show high training-MSE and low R^2 . Among DRTs, the medium tree shows better performance and medium Gaussian followed by cubic SVM out-performance other variants of SVM. Subsequently, it is interesting to note that compare to other techniques, GPR models show better performance, among which GPR construed with squared exponential kernel shows the best accuracy, while the other kernel functions show the similar accuracy. Similar behaviour was observed when these techniques were used for the construction of g_1 and g_3 , and GPR with squared exponential kernel showed the highest performance in term of training-MSE and R^2 .

D. Optimisation

To optimise the baseline hull, we utilised the Jaya Algorithm (JA) [33], which is a recently proposed simple yet effective meta-heuristic optimisation technique, whose performance has been proven in various engineering applications. JA is a population-based technique, which requires a set of randomly sampled initial solutions to start the optimisation. Unlike most of the population-based meta-heuristic optimisation techniques, JA does not require the tuning of any algorithmic specific parameters. This nature lifts an additional burden from the user and ensures a solution closer to the global optimal.

The optimisation is run in connection with the trained surrogate models as described in Section II.F, with the aim to reduced model-scale calm-water wave resistance coefficient, c_w , of the baseline ship model. As JA is a stochastic meta-heuristic technique, which may provide different results in each run, therefore 100 different optimisation runs were performed and in each run, a total of 1500 iterations were conducted. Figure 15 (a) shows the average values of c_w in 100 runs and Figure 15 (b) shows the c_w value in a single run versus the first 50 iterations performed in \mathcal{U}_1 , \mathcal{U}_2 and \mathcal{U}_3 . The optimisation results obtained after exploration of \mathcal{U}_1 , \mathcal{U}_2 and \mathcal{U}_3 are shown in Table 3.

It is noteworthy that \mathcal{U}_2 gives the most optimal design followed by \mathcal{U}_1 and \mathcal{U}_3 . Interestingly, despite showing



MLR: Multiple Linear Regression
F-Tree: Fine Regression Tree
M-Tree: Medium Regression Tree
C-Tree: Coarse Regression Tree
L-SVM: Linear SVM
Q-SVM: Quadratic SVM
C-SVM: Cubic SVM
FG-SVM: Fine Gaussian SVM
MG-SVM: Medium Gaussian SVM
CG-SVM: Coarse Gaussian SVM
Bo-Trees: Boosted Trees
Ba-Trees: Bagged Trees
Q-GPR: Quadratic GPR
M-GPR: Matern 5/2 GPR
E-GPR: Exponential GPR
SE-GPR: Squared Exponential GPR

Fig. 14 Comparison of different techniques used for the construction of g_2 .

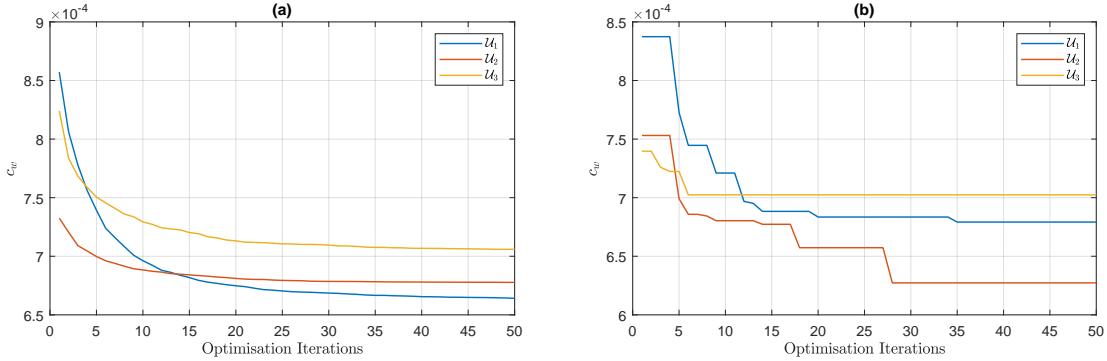


Fig. 15 Plots for the objective function (c_w) versus number of optimisation iterations performed in functionally-active subspaces of Fig. 5. (a) Average c_w over 100 optimisation runs and (b) c_w in a single optimisation run over first 50 iterations.

Table 3 Optimisation results obtained after exploring \mathcal{U}_1 , \mathcal{U}_2 and \mathcal{U}_3 in connection with g_1 , g_2 and g_3 .

	\mathcal{U}_1	\mathcal{U}_2	\mathcal{U}_3
Dimensions	5	4	1
c_w of optimised designs	0.00067353	0.00060254	0.00070930

higher accuracy both in term of dimensionality reduction and surrogate modelling, \mathcal{U}_3 does not give the most optimal design. This is probably because the geometric variability captured by this subspace is very small and has over learnt the variability of c_w . Therefore, the optimiser cannot explore diverse designs for global optimum compare to \mathcal{U}_1 and \mathcal{U}_2 . Figure 16 (a), (b) and (c) shows the shape difference between the baseline design and optimised designs obtained, respectively, from \mathcal{U}_1 , \mathcal{U}_2 and \mathcal{U}_3 . Table 4 shows the design and hydrostatic properties of the baseline and optimal designs obtained from these design spaces. Among these properties, during the shape optimisation, designers intend to

Table 4 Design and hydrostatic properties of the baseline and optimal hull forms obtained from \mathcal{U}_1 , \mathcal{U}_2 and \mathcal{U}_3 .

Parameter	Original Design	\mathcal{U}_1	\mathcal{U}_2	\mathcal{U}_3	Parameter	Original	\mathcal{U}_1	\mathcal{U}_2	\mathcal{U}_3
T	6.17000	6.17000	6.17000	6.17000	LCF	77.8780	77.4501	79.0769	78.6429
∇	2974.23	3129.98	3193.92	3048.54	A_w	2095.02	2208.29	2292.37	2153.80
S	8419.32	8425.02	8432.30	8429.33	A_m	96.0803	106.271	100.510	94.2810
Lwl	142.050	142.050	142.050	142.050	C_p	0.61688	0.55810	0.59059	0.62940
Bwl	19.0810	21.0200	20.1320	19.8810	C_m	0.81610	0.81940	0.80917	0.76864
B	20.5390	20.2057	19.8714	20.4606	C_w	0.77294	0.73957	0.80160	0.76260
LCB	71.8693	71.8677	72.4424	71.5878	C_b	0.50344	0.45731	0.47789	0.48378
VCB	2.48766	2.36010	2.38000	2.44797					

Where T : Draft, ∇ : Volume displacement, S : Surface area, Lwl : Length at Waterline, Bwl : Beam at Waterline, B : Overall beam, LCB : Longitudinal centre of buoyancy, VCB : Vertical centre of buoyancy, LCF : Longitudinal centre of flotation, A_w : Waterplane area, C_p : Prismatic coefficient, C_m : Midship coefficient, C_w : Waterplane coefficient, C_b : Block coefficient.

keep the volume displacement (∇), length (L), length at the waterline (Lwl), overall beam (B), beam at the waterline (Bwl) and draft (T) the same as baseline design. It can be seen from Table 4 that these parameters are close to each other with mirror variations. Although these variations can be reduced with design constraints during optimisation but they are not critical at least in the context of present work.

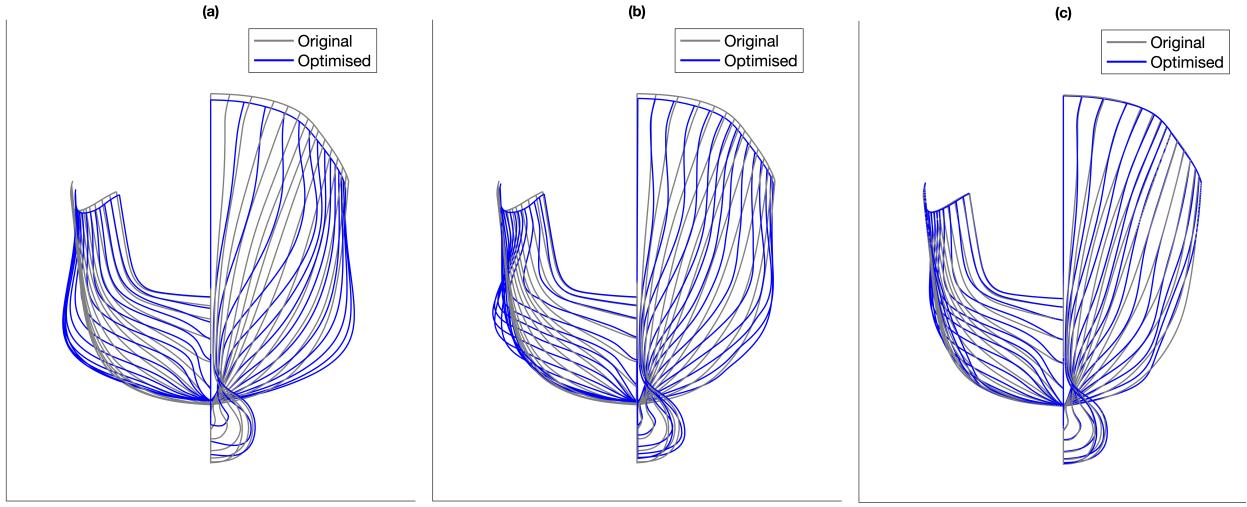


Fig. 16 Comparison between the baseline design and optimised designs obtained, respectively, from functionally-active subspaces (a) \mathcal{U}_1 , (b) \mathcal{U}_2 and (c) \mathcal{U}_3 .

In comparison to \mathcal{U}_1 , the optimised design from \mathcal{U}_2 shows a 10.54% of improvement in c_w , which is mostly associated with the sonar dome area and flat of side close to the stern. In Figure 15 (b), it is also noteworthy that optimisation performed in \mathcal{U}_2 converge faster, which is beneficial for computational cost reduction of SDO, especially when designs have to be evaluated with time expensive simulation tools during optimisation, as optimal design can be achieved with fewer design evaluations. In \mathcal{U}_3 the convergence stopped after 7th iteration.

Furthermore, as \mathcal{U}_3 is one-dimensional and \mathcal{U}_1 and \mathcal{U}_2 are five and four-dimensional, therefore, for a balanced comparison, we set the dimensionality of \mathcal{U}_3 from one to five and see if further increasing its dimensionality can help to obtain a better result. For this, five-dimensional g_3 was developed to evaluate designs in five-dimensional version of \mathcal{U}_3 . The optimal design obtained at the end of optimisation has $c_w = 0.00070373$, which does not show any significant improvement compared to the one obtained from one-dimensional \mathcal{U}_3 . Moreover, this design still has lower performance

in comparison to the design obtained from \mathcal{U}_1 and \mathcal{U}_2 .

We also tested one-dimensional behaviour of \mathcal{U}_1 and \mathcal{U}_2 . For this, the dimensionality of \mathcal{U}_1 and \mathcal{U}_2 was set to one and one-dimensional g_1 and g_2 were constructed. The optimal designs obtained from the exploration of one-dimensional \mathcal{U}_1 and \mathcal{U}_2 have c_w values of 0.001025 and 0.001004, respectively, which are higher than the one obtained from one-dimensional \mathcal{U}_3 . This shows that \mathcal{U}_3 captures well the monotonic behaviour of the problem. However, in comparison to \mathcal{U}_1 , the one-dimensional \mathcal{U}_2 shows better performance.

The efficiency of one-dimensional g_1 and g_2 is not the same as in the higher dimensions. Therefore, g_1 and g_2 were constructed again in their original dimensionality, which was five and four, respectively. Then to test the geometric and functional variability captured by the first latent variables, we again explored one-dimensional \mathcal{U}_1 and \mathcal{U}_2 with optimisation, however, during this exploration designs were evaluated with five- and four-dimensional g_1 and g_2 as they have better prediction capability than their construction in a single dimension. This optimisation problem is formulated as in Eq. (49).

$$\begin{aligned}
& \text{given } \mathbf{u}' = \mathbf{W}_1^{f'} \mathbf{v} \in \mathcal{U}', \mathbf{W}_1^{f'} \in \mathbb{R}^{N \times P'} \\
& \mathbf{u}'' = \mathbf{W}_1^{f''} \mathbf{v} \in \mathcal{U}'', \mathbf{W}_1^{f''} \in \mathbb{R}^{N \times P''} \text{ where } P' < P'' \\
& \mathbf{W}_1^{f''*} \in \mathbb{R}^{P'' \times P'} \\
& c_w \approx g'(\mathbf{u}'') \\
& \min_{\mathbf{u}'} g'(\mathbf{W}_1^{f''*} \mathbf{u}') \\
& \text{subject to } \mathbf{u}^l \leq \mathbf{u}' \leq \mathbf{u}^u \\
& \quad \mathbf{u}^l \leq \mathbf{W}_1^{f''*} \mathbf{u}' \leq \mathbf{u}^u \\
& \text{yield } \mathbf{u}'_{optimal}
\end{aligned} \tag{49}$$

Here, \mathcal{U}' is the one-dimensional ($P' = 1$) version of \mathcal{U}_1 and \mathcal{U}_2 , and \mathcal{U}'' represents their original dimensionality (i.e., P'' equal to four for \mathcal{U}_1 and equals to five for \mathcal{U}_2). During exploration of \mathcal{U}' for optimal design, $\mathbf{u}'_{optimal}$, $\mathbf{W}_1^{f''*}$ projects the lower-dimensional design, $\mathbf{u}' \in \mathcal{U}'$, on the higher-dimensional space \mathcal{U}'' , which is evaluated with surrogate model, g' , to guide the optimiser. Here, g' is trained in \mathcal{U}'' thus has the dimensionality of P'' . Moreover, the first constraint in Eq. (49) defines the viable search space for exploration of $\mathbf{u}'_{optimal}$. Whereas second constrain in Eq. (49) ensures that design evaluated with g' lies within its domain on which this model was initially trained. As the evaluation of design's c_w , which lie outside \mathcal{U}'' will results in its false/inaccurate estimation, resulting in guiding the optimisation towards local optima.

Table 5 Summary of the optimisation results under different dimensionality of subspaces and surrogate models.

	\mathcal{U}_1	g_1	\mathcal{U}_2	g_2	\mathcal{U}_3	g_3
Dimensions	5	5	4	4	1	1
c_w	0.0006735		0.0006025		0.00070373	
Dimensions	1	1	1	1	5	5
c_w	0.0010250		0.0010040		0.00070373	
Dimensions	1	5	1	4		
c_w	0.0008759		0.0008506			

In this comparison, the optimal designs obtained from one-dimensional \mathcal{U}_1 and \mathcal{U}_2 have c_w equal to 0.00087596 and 0.00085064, respectively. This concludes that evaluation of designs with five- and four-dimensional g_1 and g_2 during the exploration of one-dimensional \mathcal{U}_1 and \mathcal{U}_2 shows better results compare to the evaluation of designs with one-dimensional g_1 and g_2 . Even in this case, \mathcal{U}_2 provides the most optimal design and the monotonic behaviour of the problem is well defined with \mathcal{U}_3 . Table 5 gives the summary of all the optimisation test performed under different dimensionality of subspaces and surrogate models. These results again demonstrate that the dimensionality reduction performed with feature-to-feature learning encodes well the lower-dimensional latent subspace compared to the one performed with signal-step feature extraction.

V. Conclusions & Future works

This paper proposed a dimensionality-reduction method to reduce the computational cost and increase the efficiency of SDO. The proposed approach commences by feature extraction to learn a lower-dimensional latent space in which basis of the original design space was formed while preserving the maximum geometric variability of the designs. Afterwards, another set of features of this subspace were extracted to further reduced its dimensional. This two-stage feature-to-feature learning creates a functionally-active subspace while retaining the geometric and functional variability of original design space. The geometric features help optimisation to explore diverse design in a lower dimension and functional features facilitates the efficient surrogate model training. We tested the proposed approach on a 27-dimensional design space, which was formulated for optimisation of DTMB 5415 ship hull. The objective for optimisation was to minimise the calm-water resistance (c_w), where c_w was evaluated using the surrogate model trained in the functionally-active subspace. Three different methodological pipelines were tested and results showed that compared to conventional feature learning techniques, the two-step feature extraction provides beneficial improvement to the computational efficiency of SDO.

As a future work, along with the design parameters, the effect of geometry-based shape descriptor on feature extraction will be studied. Authors are also keen to work on the development of a Psycho-Physical metric [34] to visually access the geometric variance retained by a certain subspace, which might facility designers to effectively explore subspaces in generative design paradigm.

Acknowledgements

The first author is grateful to the Mac Robertson Trust for sponsoring his visit to CNR-INM through their Postgraduate Travel Scholarship program. CNR-INM authors are grateful to the US Office of Naval Research for its support through NICOP grant N62909-18-1-2033. The first and last author of this work has also received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant GRAPES (agreement No. 860843).

References

- [1] Schulz, A., Xu, J., Zhu, B., Zheng, C., Grinspun, E., and Matusik, W., "Interactive design space exploration and optimization for CAD models," *ACM Transactions on Graphics (TOG)*, Vol. 36, No. 4, 2017, pp. 1–14.
- [2] Khan, S., Gunpinar, E., and Dogan, K. M., "A novel design framework for generation and parametric modification of yacht hull surfaces," *Ocean Engineering*, Vol. 136, 2017, pp. 243–259.
- [3] Khan, S., and Gunpinar, E., "Sampling CAD models via an extended teaching–learning-based optimization technique," *Computer-Aided Design*, Vol. 100, 2018, pp. 52–67.
- [4] D'Agostino, D., Serani, A., and Diez, M., "Design-space assessment and dimensionality reduction: An off-line method for shape reparameterization in simulation-based optimization," *Ocean Engineering*, Vol. 197, 2020, p. 106852.
- [5] Chen, W., Fuge, M., and Chazan, J., "Design manifolds capture the intrinsic complexity and dimension of design spaces," *Journal of Mechanical Design*, Vol. 139, No. 5, 2017.
- [6] Gunpinar, E., and Khan, S., "A multi-criteria based selection method using non-dominated sorting for genetic algorithm based design," *Optimization and Engineering*, Vol. 21, No. 4, 2020, pp. 1319–1357.
- [7] Chen, W., and Fuge, M., "Synthesizing designs with interpart dependencies using hierarchical generative adversarial networks," *Journal of Mechanical Design*, Vol. 141, No. 11, 2019.
- [8] Gunpinar, E., Coskun, U. C., Ozsipahi, M., and Gunpinar, S., "A generative design and drag coefficient prediction system for sedan car side silhouettes based on computational fluid dynamics," *Computer-Aided Design*, Vol. 111, 2019, pp. 65–79.
- [9] Diez, M., Campana, E. F., and Stern, F., "Design-space dimensionality reduction in shape optimization by Karhunen–Loève expansion," *Computer Methods in Applied Mechanics and Engineering*, Vol. 283, 2015, pp. 1525–1544.
- [10] Burnap, A., Pan, Y., Liu, Y., Ren, Y., Lee, H., Gonzalez, R., and Papalambros, P. Y., "Improving design preference prediction accuracy using feature learning," *Journal of Mechanical Design*, Vol. 138, No. 7, 2016.
- [11] Yonekura, K., and Watanabe, O., "A shape parameterization method using principal component analysis in applications to parametric shape optimization," *Journal of Mechanical Design*, Vol. 136, No. 12, 2014.

- [12] D'Agostino, D., Serani, A., Campana, E. F., and Diez, M., "Nonlinear methods for design-space dimensionality reduction in shape optimization," *International Workshop on Machine Learning, Optimization, and Big Data*, Springer, 2017, pp. 121–132.
- [13] D'Agostino, D., Serani, A., Campana, E. F., and Diez, M., "Deep autoencoder for off-line design-space dimensionality reduction in shape optimization," *2018 AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, 2018, p. 1648.
- [14] Khan, S., Gunpinar, E., and Sener, B., "GenYacht: An interactive generative design system for computer-aided yacht hull design," *Ocean Engineering*, Vol. 191, 2019, p. 106462.
- [15] Serani, A., and Diez, M., "Shape Optimization under Stochastic Conditions by Design-space Augmented Dimensionality Reduction," *19th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference (MA&O), AVIATION 2018*, Atlanta, GA, USA, June 25-29, 2018.
- [16] Serani, A., D'Agostino, D., Campana, E. F., and Diez, M., "Assessing the Interplay of Shape and Physical Parameters by Unsupervised Nonlinear Dimensionality Reduction Methods," *Journal of Ship Research*, 2019.
- [17] Lukaczyk, T. W., Constantine, P., Palacios, F., and Alonso, J. J., "Active subspaces for shape optimization," *10th AIAA multidisciplinary design optimization conference*, 2014, p. 1171.
- [18] Constantine, P. G., *Active subspaces: Emerging ideas for dimension reduction in parameter studies*, Vol. 2, SIAM, 2015.
- [19] Williams, C. K., and Rasmussen, C. E., *Gaussian processes for machine learning*, Vol. 2, MIT press Cambridge, MA, 2006.
- [20] Schulz, E., Speekenbrink, M., and Krause, A., "A tutorial on Gaussian process regression: Modelling, exploring, and exploiting functions," *Journal of Mathematical Psychology*, Vol. 85, 2018, pp. 1–16.
- [21] Umetani, N., and Bickel, B., "Learning three-dimensional flow for interactive aerodynamic design," *ACM Transactions on Graphics (TOG)*, Vol. 37, No. 4, 2018, pp. 1–10.
- [22] Snoek, J., Larochelle, H., and Adams, R. P., "Practical bayesian optimization of machine learning algorithms," *Advances in neural information processing systems*, 2012, pp. 2951–2959.
- [23] Grigoropoulos, G., Campana, E., Diez, M., Serani, A., Goren, O., Sarioz, K., Danisman, D., Visonneau, M., Queutey, P., Abdel-Maksoud, M., et al., "Mission-based hull-form and propeller optimization of a transom stern destroyer for best performance in the sea environment," *Proceedings of the VII International Congress on Computational Methods in Marine Engineering-MARINE*, 2017.
- [24] Serani, A., Diez, M., Wackers, J., Visonneau, M., and Stern, F., "Stochastic Shape Optimization via Design-Space Augmented Dimensionality Reduction and RANS Computations," *57th AIAA Aerospace Sciences Meeting, SciTech 2019*, Manchester Grand Hyatt San Diego, San Diego, California, January 7-11, 2019.
- [25] Dawson, C. W., "A practical computer method for solving ship-wave problems," *Proceedings of the 2nd International Conference on Numerical Ship Hydrodynamics*, Berkeley, 1977, pp. 30–38.
- [26] Schlichting, H., and Gersten, K., *Boundary-Layer Theory*, Springer-Verlag, Berlin, 2000.
- [27] Bassanini, P., Bulgarelli, U., Campana, E. F., and Lalli, F., "The wave resistance problem in a boundary integral formulation," *Surveys on Mathematics for Industry*, Vol. 4, 1994, pp. 151–194.
- [28] Pastor, M., Bindu, M., and Harčarik, T., "Modal assurance criterion," *Procedia Engineering*, Vol. 48, 2012, pp. 543–548.
- [29] Othmer, C., Lukaczyk, T. W., Constantine, P., and Alonso, J. J., "On active subspaces in car aerodynamics," *17th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, 2016, p. 4294.
- [30] Constantine, P., and Gleich, D., "Computing active subspaces with Monte Carlo," *arXiv preprint arXiv:1408.0545*, 2014.
- [31] Constantine, P. G., and Diaz, P., "Global sensitivity metrics from active subspaces," *Reliability Engineering & System Safety*, Vol. 162, 2017, pp. 1–13.
- [32] Lodhia, Z., Rasool, A., and Hajela, G., "A survey on machine learning and outlier detection techniques," *International Journal of Computer Science and Network Security*, Vol. 17, No. 5, 2017, pp. 271–276.
- [33] Rao, R., "Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems," *International Journal of Industrial Engineering Computations*, Vol. 7, No. 1, 2016, pp. 19–34.
- [34] Khan, S., Gunpinar, E., Moriguchi, M., and Suzuki, H., "Evolving a psycho-physical distance metric for generative design exploration of diverse shapes," *Journal of Mechanical Design*, Vol. 141, No. 11, 2019.