IDEAS@MI: Intelligent Data-driven systEms for digitAl deSign in Maritime Industry

Shahroz Khan

CAD/CAx Research Group

Department of Naval Architecture, Ocean and Marine Engineering

University of Strathclyde, Glasgow

August 16, 2023

This thesis is the result of the author's original research. It has been composed by the author and has not been previously submitted for examination which has led to the award of a degree.

Abstract

Aligning maritime design schemes with Industry-4.0 and -5.0 trends, this PhD thesis aims to propel initiatives for developing novel data-driven technologies that cover the full spectrum of simulation-driven design optimisation activities by i) improving the efficiency of design space exploration, ii) reducing the overall computational cost, iii) developing versatile design parameterisation, and iv) integrating human intelligence in the design process. These objectives are achieved by proposing new novel tools and techniques within parametric sensitivity analysis (PSA) and feature extraction paradigms to eliminate less significant towards the designs' physics and construct geometry-driven, physics-informed and user-integrated subspaces.

First, a novel intra-sensitivity concept is proposed to study the local behaviour of parametric sensitivities and eliminate instabilities - a parameter can be sensitive in certain local areas of the design space but become insensitive in others. Therefore, the outcome of intra-sensitivity allows designers to construct viable design spaces for the reliable execution of PSA. Afterwards, implementation of PSA or intra-sensitivity is expedited with a new geometric-moment dependent PSA that harnesses the geometric variation in a design space using geometric moments to measure parametric sensitivities. A shape-supervised dimension reduction approach is also developed. It extracts a high-level geometry description as a shape signature vector and uses it as a substitute for physics to construct a physics-informed design subspace. A feature-to-feature learning strategy is also proposed to create a functionally-active subspace for expediting the construction of surrogate models

at an off-line stage. For the versatile parameterisation of ship hulls, we developed ShipGAN using deep convolutional generative adversarial networks, so the resulting parametric modeller is generic with the ability to perform feasible and plausible design modifications for a large variety of hulls. Finally, we propose a generative and interactive design tool which aids users during optimisation by guiding the design exploration towards user-centred and physically optimised designs.

Contents

List of Figures

List of Tables

Acknowledgements

"No one who achieves success does so without acknowledging the help of others. The wise and confident acknowledge this help with gratitude."

Alfred North Whitehead

First of all, I would like to thank my thesis advisor, Prof. Panagiotis Kaklis and co-advisor, Prof. Fehim Cirak, from the University of Cambridge, for providing me with the scientific grounding coupled with the legitimacy to complete this thesis with the same scientific aims I started my PhD - a digital tool for the maritime industry to generate appearance- and performance-driven designs at the conceptual design phase and to overcome the computational burden of physical computer simulations.

I owe my deepest and most sincere gratitude to Prof. Kaklis for his utmost support in every stage of my PhD journey. I thank him for infusing a creative aspect into my scientific research and helping me realise the fundamentals' value. This provided me with concrete pillars of computational geometry to structure my research and propose something completely novel. Through his patience and perseverance, he tasked me to implement concepts and methods from data-driven science to the domain of maritime design. His insightful suggestions and planning strategies have made all of this achievable in good time with an upbeat spirit. Thank you for your warmth and your guidance, Prof. Kaklis.

I joined Strathclyde's department of naval architecture, ocean and marine engineering (NAOME) as a *Marie Skłodowska-Curie fellow* within the "GRAPES (learninG, pRocessing And oPtimising shapES)", which received funding from the European Unions Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 860843. It was the EU's this generous support that enabled me to attend various national and international conferences and events (15 in total) and showcase my work to a broader audience. More importantly, this also allowed me to emphasise the importance of computational creativity to the maritime community.

I would like to thank the Dean, Prof. Atilla Incecik, for supporting my application to the department. For giving me his valuable time for long, yet exciting, chats during departmental events and encouraging me to work harder.

This thesis would not be in its present form without my various research visits at the Virtual Reality Research Group of the Ship Design Laboratory at the School of Naval Architecture and Marine Engineering (S-NA&ME), National Technical University of Athens (NTUA), Athens, Greece; Institute of Marine Engineering of the Italian National Research Council (CNR-INM), Rome, Italy; Co-Design Lab, Berkeley Institute of Design, University of California, Berkeley, US; and more recently at ITI-Global, Cambridge, UK.

Very special thanks I shall extend to Dr Matteo Diez and Dr Andrea Serani from CNR-INM for accepting me as a *Mac Robertson visiting fellow* and welcoming me to their research group. Without their fruitful collaboration, much of the scientific contribution in this thesis would not be possible. This visit would not be possible without the Mac Robertson trust's generous support. Therefore, I owe a special thanks to Mac Robertson Trust. I am also grateful to Prof. Kosa Goucher-Lambert from the Co-Design lab, to whom I am in debt for creative and intellectual stimulation, thereby letting me explore the Design Theory and Methodology aspect of my work. I would also like to thank colleagues at the Co-Design lab for their invaluable friendship and support and for cultivating a discourse from the bottom up informed by designing first and theorising later. The visit to the Co-Design lab was made possible with the support from *the Royal Society*, who funded

this visit under the HINGE (Human InteractioN supported Generative modEls for creative designs) project via their "International Exchanges 2021 Round 2" funding call. I would like to give special thanks to Prof. Alexandros-Alvertos I. Ginnis from NTUA for hosting me in his virtual reality research group, in which I had a unique opportunity to study how one could couple parametric design tools with virtual reality systems to develop an immersed computational design platform for complex maritime vessels.

I would like to acknowledge insightful and influential conversations with Prof. Konstantinos Kostas from Nazarbayev University and Prof. Constantinos Politis from the University of West Attica during our weekly meetings. I would like to thank our collaborators from ITI, Cambridge, UK; Monolith AI, London, UK; CAE-Tech, Leamington Spa, UK; Institute of Intelligent Industrial Technologies and Systems for Advanced Manufacturing (STIIMA-CNR), Italy; Israel Institute of Technology, Haifa, Israel and Istanbul Technical University, Istanbul, Turkey.

I would like to express special gratitude towards my good friend and colleague, Ondar, Alex, Abdullahi and Yildirim, from the NAOME department, with who I shared countless innovative ideas, great coffee chats and delicious meals. Within the NAOME department, I would also like to thank the incredible administrative team, Susan, Ross, and Lynne, for being patient with me and supporting me in every administrative work.

I want to acknowledge my mentors from my undergraduate and Master's studies in Pakistan, Turkey, and Japan. Though they were not directly involved in this thesis, they were fundamental in preparing me for it. I am very grateful for their teaching, mentorship, and support.

A warm expression of gratitude to my best friends, Muhammad Saqib Jabar, Imtiaz Ahmed Qurashi, Nokukhanya Tsabedze and Dilan Könes, who have been to me like a secret weapon to get through the hard times with love and emotional endurance that are rare to find.

And finally, to my siblings, Sanjeeda Andlib, Shahzad Akhtar Khan, Farida Nadeem, Aneela Ashraf, Sajjad Hussain Akhtar, Jahanzaib Barakzai, Sahrish Khan, and my parents,

Chapter 0. Acknowledgements

Muhammad Akhtar Khan and Saeeda Akhtar, for supporting me in every step I took and decision I made in life. My mother, who, with her elegance, has made my life's journey worthwhile. She taught me everything I know of the world's ways and continues to move and surprise me with her awe-inspiring wisdom and divine grace. I owe every bit of success to my father for teaching me that life is not worth living without taking risks. He let me grow as an individual and helped me take the road less travelled.

To my beloved parents.

Chapter 0. Acknowledgements

Chapter 1

Introduction

"*The digital revolution is far more significant than the invention of writing or even of printing.*"

Douglas Engelbart

With the rapid technological advancements and worsening environmental conditions, there has been an evergrowing demand for sustainable and innovative design solutions optimised against safety, performance, reduced cost and enhanced visual aesthetics. A typical product design process involves a series of nontrivial design phases, among which the conceptual phase is recognised as of foundational and fundamental importance. Designers need to explore various creative and ingenious alternatives, which are initially executed via two-dimensional (2D) sketches followed by three-dimensional (3D) computer-aided design (CAD) models and physical prototypes for validation within designs' functional requirements and target customers' preferences. However, this phase can be time-consuming, especially if the products' functionality, practical validity and visual appearance are all critical drivers of the design process. This is mainly because the design may be functionally optimised but impractical from a useability and manufacturing point of view or does not meet customers' psychological preference for the product's appearance.

Therefore, it is essential to have a robust computational design tool to facilitate designers and engineers throughout product development to synthesise a diverse set of design ideas driven by engineers' performance requirements and designers' perceptions regarding designs' feasibility and form appearance. More importantly, such a tool will also ensure economic product development, allowing designers and engineers sufficient time to focus on its sustainability and environmental, social, and ethical impact. As for many products, their early stages of development have been identified as the stages most adversely affecting the environment.

The evolution of computational power has enabled designers and engineers to expedite traditional product design with computer-aided design (CAD) tools and physics-based simulations, which have become indispensable for handling various design problems. Their involvement throughout this process engendered *simulation-driven design (SDD)*, which involves rapid exploration of parametric design spaces for global optima leading to shorter product development cycles. Typically, SDD commences with an intuitive design parametrisation for formulating diverse and rich design/search spaces. These spaces are explored in conjunction with optimisers and shape modification methods to improve a baseline design based on certain performance criteria. See Fig.1.1 for the overall workflow of a typical SDD pipeline.

For design problems involving free-form shapes, SDD often suffers from high computational costs, which are associated mainly with the

1. parametrisation of a baseline design,
2. its physics evaluation [5] and
3. design space exploration.

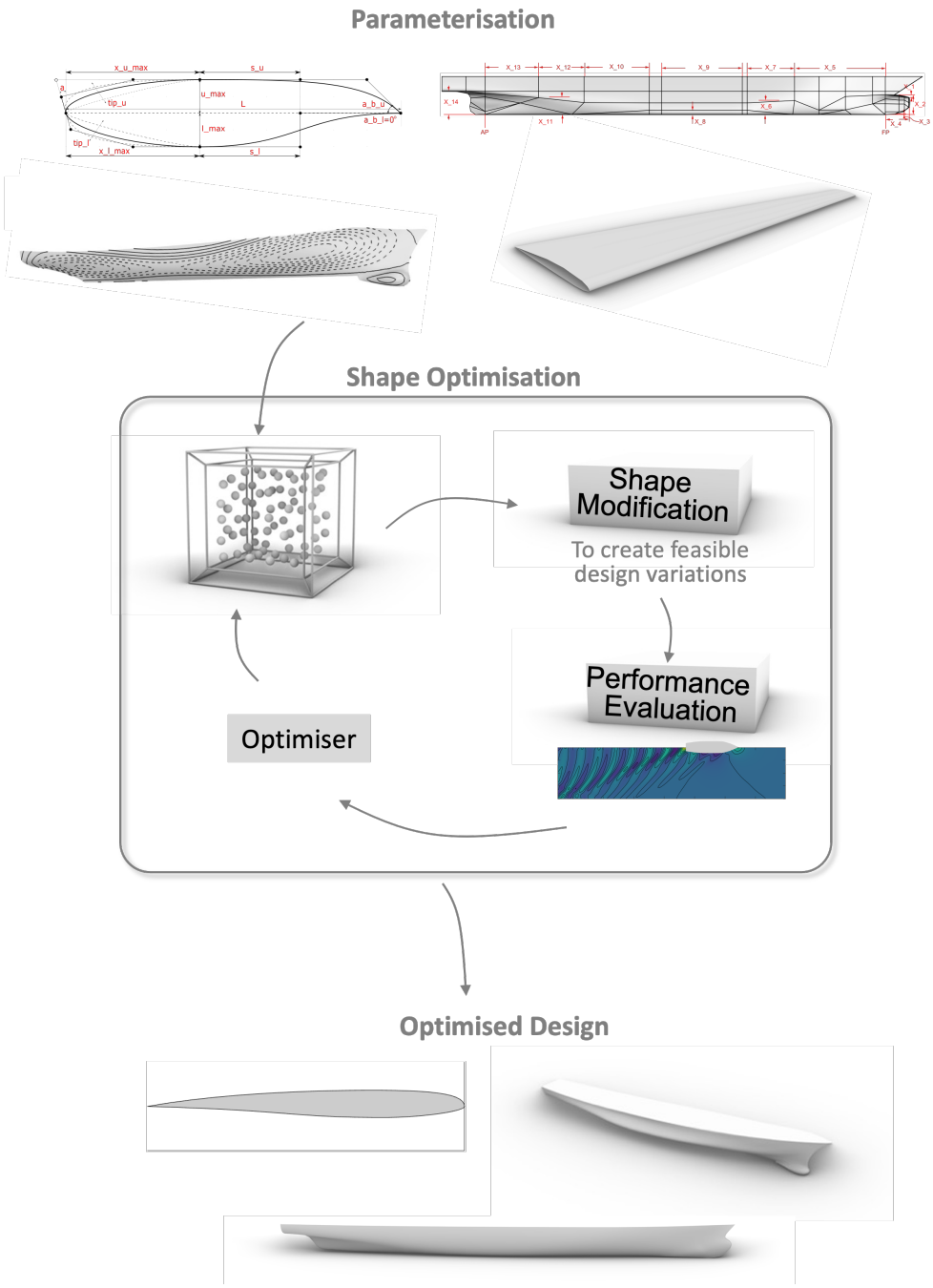


Figure 1.1: Workflow of typical simulation-driven shape optimisation.

1.1 Effective design parameterisation

In parametrisation, such a cost is initially steered by the type of parametrisation conducted and the number of design parameters used. The number of these parameters has an exponential impact on the overall computational cost, thereby giving rise to the curse of dimensionality [6]. Compared to low-dimensional design spaces, a high-dimensional one may increase the drive towards a globally optimal design, but it will be at the expense of exhaustive exploration resulting in an evaluation of numerous designs involving computationally demanding simulations. Therefore, a practical parametrisation should be sufficiently versatile to cover a large spectrum of design possibilities and concise enough to parametrise all critical features with only a few parameters [7, 8] (see Fig. 1.2 for an example of a parametric design).

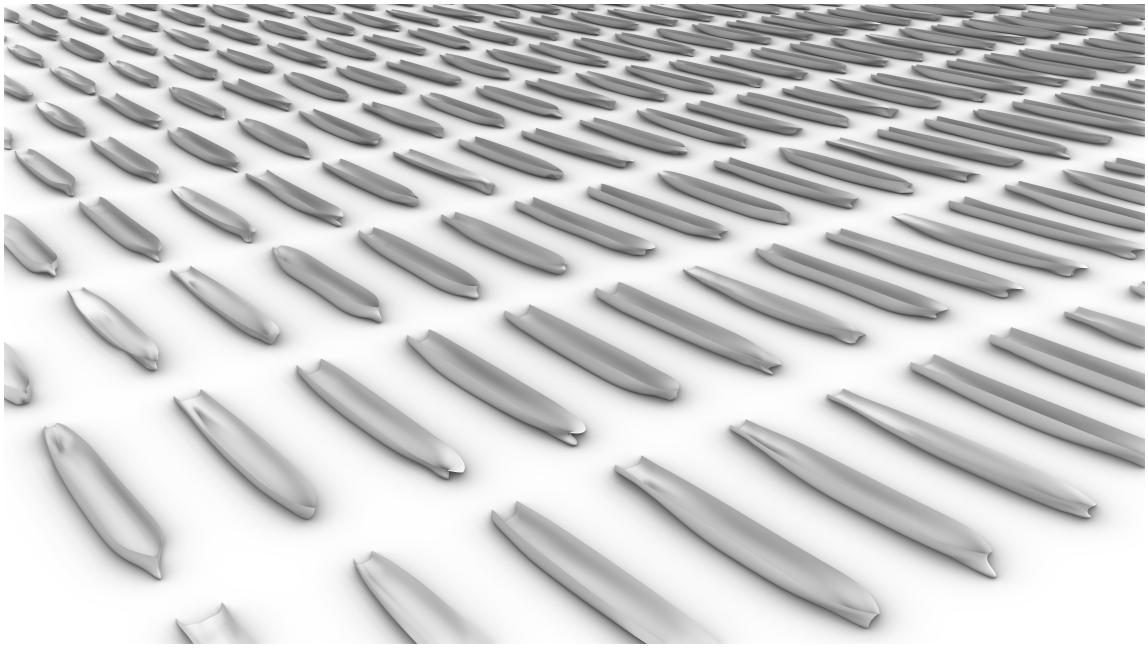


Figure 1.2: An example of parametric design modification of ship hulls.

Therefore, the design parameterisation tasks can be extensively tricky. The quality of design spaces used in SDD substantially impacts their outcome. These spaces are usually

narrow, with minimal design variations and may contain invalid designs. More importantly, these spaces are solely constructed based on the design geometry with no information on physics.

In SDD, performance evaluation of designs is conducted with physics-based simulation tools, such as Computational Fluid Dynamics (CFD), Computational Structural Dynamics (CSD), Computation Aeroacoustics (CAA), etc., which are the key drivers of design space exploration [9], and are continuously evolving to achieve a high level of fidelity. However, simultaneously they are becoming computationally intensive, requiring unaffordable computational resources even for a single simulation run [10]. As an example, a CFD-based statistically significant evaluation of ship performance in waves may require up to 1M CPU hours on HPC systems [11]. Consequently, the extensive use of these tools can be impractical for rich and vast design spaces, which are often incurred by design constraints to confine exploration to feasible designs [12]. A more critical bottleneck can be encountered for high-dimensional design spaces, which, as described earlier, are favoured for maximal performance improvement [13]. This further ignites the curse of dimensionality, thus hampering optimisation's success.

1.2 Tackling computational complexity

The existing techniques used to reduce the exorbitant computational costs mainly fall into two categories. One line of work focuses on developing computationally less demanding solvers [14, 15] while the other leverages computational resources with data-driven approaches [16, 17]. Recently, the most astounding results in reducing computational cost while maintaining the high approximation accuracy of designs' physics are achieved via data-driven approaches. These approaches can be broadly classified as dimensional reduction and surrogate modelling, employed to reduce the design space dimensionality and bypass designs' performance estimation with simulation tools, respectively [18].

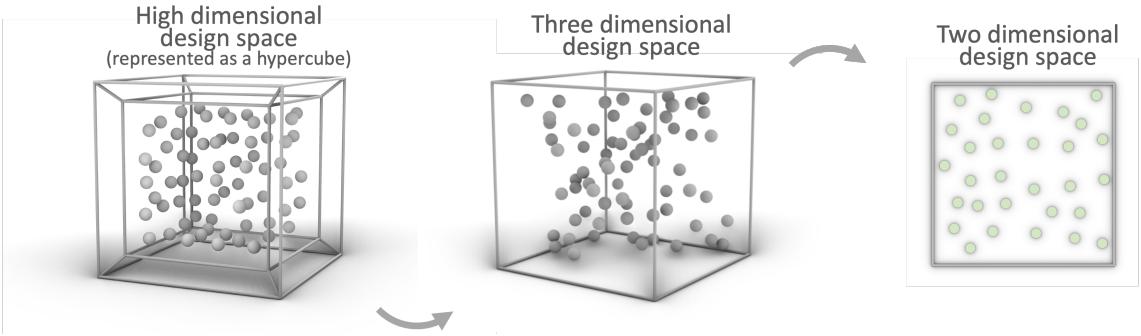


Figure 1.3: The dimensionality of the original high-dimensional design space (represented here as a hypercube) can be reduced significantly by eliminating the parameters less sensitive to the physical quantity of interest or by extracting the latent features along which materialise maximum variation in the geometry. The resulting lower-dimensional space can be used for efficient design exploration, thereby expediting the optimisation and potentially cutting down the overall computational cost.

1.2.1 Design space dimensionality reduction (DSDR)

The common cure is dimensionality reduction, for which there are well-studied unsupervised [12, 19] and supervised [20] approaches (see Fig. 1.3 for an illustrative example of DSDR). Implementing supervised techniques require consideration of design physics, but using unsupervised ones does not. Unsupervised approaches are also referred to as feature extraction (FE) or manifold learning. These approaches aim to extract latent features/variables from the design space, which can be classified as geometrically active or inactive depending on their importance in affecting a shape's geometric variability [21]. Inactive features are redundant, and their usage has no or minimal impact on shape variation and performance improvement during optimisation; thus, they can be safely ignored to reduce the space's dimensionality. The geometrically active latent features form a new set of parameters for shape modification and construct a basis spanning a lower-dimensional subspace for faster optimisation convergence with fewer computationally intensive design evaluations.

The widely used FE-based design space dimensionality reduction (DSDR) approaches

include Active Subspace Method (ASM) [20], Karhunen-Loève Decomposition (KLD) [22] (closely related to Principal Component Analysis (PCA), the so-called proper orthogonal decomposition [12, 16]) and their non-linear extensions, such as kernel PCA [23], ISOMAP [24], LLE [25] to handle design space non-linearities if present. Recently, Machine Learning-based approaches, autoencoders [23,26] and their variations [27,28], emerged from applications in image analysis, object recognition, speech analysis, clustering, and data visualisation etc., have gained attention in DSDR literature.

In contrast to the above-described FE-based approaches, DSDR is also performed with a parametric sensitivity analysis (PSA) [18], which are supervised approach and measures the sensitivity of each parameter towards the variability of a design's physical performance. The degree of sensitivity is measured with a metric called the *sensitivity index*, which facilitates parameters' ranking. As less sensitive parameters affect a shape's physical performance less, they can be excluded to reduce a problem's overall dimensionality.

1.2.2 Unsupervised (FE) versus supervised (PSA) DSDR

The efficiency of both supervised and unsupervised DSDR approaches is manifested in various applications to mitigate the curse of dimensionality [19,21,29]. However, since FE-based unsupervised techniques do not require performance labels, their implementation can be less expensive than supervised methods. In contrast, The implementation of supervised DSDR such as PSA is more informed because, along with dimension reduction, its assessment is a meaningful prerequisite to reducing uncertainty [30] and identifying the driving features of designs that account for the maximum or minimum variability in performance. Through these effects, PSA advances enhanced resource allocation from the preliminary design stage, thus expediting the entire product development for maximal performance improvement [31].

Drawback of unsupervised DSDR (FE): Implementing FE-based DSDR approaches can be ineffectual, primarily when no direct correlation exists with associated shape modification [12,16]. The resulting subspace will have a basis forming merely a new orientation

of the design space without capturing any associated geometric features [12, 29]. Moreover, another deficiency of unsupervised approaches is their inability to preserve a shape's complexity and its intrinsic underlying geometric structure. Thus, the resulting subspace lacks the representation *capacity* and *compactness*, which, as defined in [28, 32], is subspace's ability to produce *diverse* and *valid* shapes, respectively, with least number of latent variables when being explored for shape optimisation. These deficiencies can hamper the success of the optimiser as it may spend the majority of the available computational budget on exploring infeasible, practically invalid and similar shapes. Therefore, subspace may not be an optimisation-efficient subspace because, even if a high geometric variation is preserved, maximum design improvements are not guaranteed; see [21, 26, 27, 33]. However, it should be noted that these techniques' inability to extract appropriate geometric or physics-associated features is not necessarily an intrinsic characteristic; it mainly stems from the geometry representations used in subspace learning, which are commonly low-level shape discretisations. Thus, extracting intrinsic latent information from such representations becomes implausible; richer representations with high-level information related to the underlying shape's structure and physics are imperative.

Drawback of supervised DSDR (PSA): The implementation of PSA can be prone to

1. intrinsic instabilities, and
2. high computational cost.

The instabilities in PSA are induced due to the fact that a parameter can be sensitive within a particular local region of the design space but become insensitive in some other areas. Therefore, setting a viable design space becomes challenging for robust and desired results. Sensitivity analysis within a non-viable design space can result in an inconsistent and false estimation of parameter sensitivity, wasting time and computational resources. Whereas the computational cost of PSA is mainly induced due to the requirement of performance labels.

Instabilities due to parametrisation and design space: From the design parameterisation point of view, these instabilities mainly stem from complexities of geometric representation. This problem is proficient in the parametrisation of free-form shapes compared with the parametrisation of solid models, whose parameters directly affect design features; free-form parametrisation is only partially driven by features [34]. This nature makes it challenging to quantify key features, which can essentially materialise the most significant improvement in design performance. Different techniques can be used to parametrise a free-form shape depending on the underlying geometric representation, but such usage may draw different PSA results. Thus, the selection of suitable parametric techniques can also be challenging, dissuading designers from adopting PSA.

At the preliminary stage, the distribution of design parameters is unknown; therefore, PSA is performed within a design space established by defining these parameters' upper and lower bounds (i.e., limits). These bounds create *parametric ranges*, over which PSA quantifies parametric sensitivity. Along with the dimensionality issue, the decision on the choice of these limits of a design space should be made tentatively. It should not be narrow so that it restricts new ideas, which is particularly important at a preliminary. At the same time, space should not be so wide as to result in the wastage of computational energy in exploring non-viable regions [35].

More importantly, variability in parametric ranges causes fluctuations in the sensitivity of design parameters obtained from PSA [36]. A parameter can be sensitive within a certain region of a design space and becomes insensitive in another. A slight perturbation in the range of any parameter may cause not only a deviation in its own sensitivity index but also in the sensitivity of the remaining ones. In this context, securing robust sensitivity results can be difficult. If inappropriate ranges are chosen, then PSA results can mislead the designer about the inherent sensitivity of parameters. Therefore, it is essential to investigate the influence of different parametric ranges on PSA results, which will be referred to as *intra-sensitivity*, defined as the process of identification of parameters whose perturbation in the range is most influential on the sensitivity of others. Once identified, intra-sensitive

parameters can be tuned further to avoid uncertainty in the sensitivity results.

Computational complexity of PSA: The analytical implementation of PSA can be complex, especially if PSA necessitates the evaluation of high-dimensional integrals or if Quantity of Interest (QoI), representing designs' certain performance criterion, cannot be evaluated analytically. Therefore, PSA is approximated with sampling techniques, which can induce computational complexities in case of high-dimensional problems with costly QoIs [37]. Though Monte Carlo (MC) sampling, which is widely used, is not much influenced by the dimensionality; however, it is susceptible to slow convergence, as it requires a sufficiently large number of samples for stable results. A slightly better convergence rate can be obtained with quasi-Monte Carlo (Q-MC) or Latin-Hypercube (LH) methods based on uniformly distributed design sequences. Nevertheless, their advantages also disappear in high-dimensional design spaces [38]. Therefore, it is essential to have a robust approach that progressively creates an optimal sample set, uniformly covering the high-dimensional spaces with few numbers of diverse samples hence approximating a solution closer to the analytical one.

If PSA is to be implemented at the preliminary design stage, then at this stage, a designer is not always interested in accurately estimating the performance, so using a meta-model or computationally less demanding solvers in PSA can result in reducing the computational cost. Therefore, to lessen the computational burden, surrogate models are often used to accelerate the PSA as, at the preliminary stage, designers may not be interested in accurately estimating the performance [37]. Surrogate modelling is also a supervised learning approach. Despite the undoubted efficiency, their usage is often hindered by the availability of data, which is profound in engineering applications where data is the outcome of expensive physical simulations [10] and may exacerbate the entire design pipeline. Recently, to combat this insufficiency, these approaches have been revitalised via *scientific machine/deep learning*, such as Physics-Informed Neural Networks (PINN) [17]. PINNs are trained to integrate differential equations modelling the physics along with a moderate amount of data from simulations or experiments to approximate the underlying

Partial Differential Equations (PDE) solution. In fact, if the PDE problem is well-defined along with appropriate initial and boundary conditions, then PINN can identify the unique solution without any simulation data [10]. Their potential has been exploited in fluid mechanics [17], solid mechanics [39], and dynamical systems [40].

1.3 Efficient and intuitive design exploration

Efficient exploration of design spaces is another challenge that SDD has to overcome. Even if the design space is well-defined, exploring and finding a global solution satisfying all design requirements can still be challenging. The exploration is conducted with suitable optimisation algorithms, which are well separated into three categories: exact, heuristic and metaheuristic techniques. Exact techniques are built on a strong mathematical foundation. Their usage is limited for complex problems in the maritime industry as they require the evaluation of complex functional derivatives of the metrics involved, which would increase the computational cost with regard to the dimension of the design space. Heuristics, on the other hand, are problem-dependent optimisers, and though computationally affordable, they cannot guarantee to provide the global optimum. Finally, metaheuristics are not-problem-dependent techniques; however, in order to avoid getting trapped near local optima, metaheuristics have to perform a high number of iterations involving the evaluation of performance metrics, which is not unlikely to be of inhibitive computational cost in the case of high-fidelity simulations, e.g., when using a CFD tool for evaluating the total resistance of a ship in full scale for a single Froude number.

Furthermore, the non-intuitive nature of these techniques cannot capture the designers' design intention as the optimisers solely drive the optimisation based on physics. The resulting design may be optimal for specific performance criteria but infeasible for practical usage. Therefore, during the exploration of design space, optimisers should not only find the optimal design in terms of physical performance but also a design which captures the designers' perception related to its physical form. As it is critical to generate a fi-

nal shape that, along with the design specifications, meets the customers' physiological preference for the product's appearance to make it successful within the market, e.g., the physical form or the layout that would render a new yacht design successful in the market. Therefore, an essential element of the design exploration process should be the inclusion of psychological aspects that affect the design. This is especially crucial when it comes to the optimisation of free-form shapes like ships. Although some academic scholars from the maritime field have made a considerable contribution to the modernisation of preliminary ship design techniques, their usage in the industry is still limited. Some of the recent efforts to support ship design at the initial stage include the development of attribute-based design techniques [41]; parametric design systems [42]; library-based [43], sketching based [44], interactive optimisation [45] based and three-dimensional packing based [46,47] approaches for exploration of hull form variations; simulation-driven [48] and holistic approach to ship design [49] and machine learning-based ship design method to assist the optimisation towards the optimal solution [50].

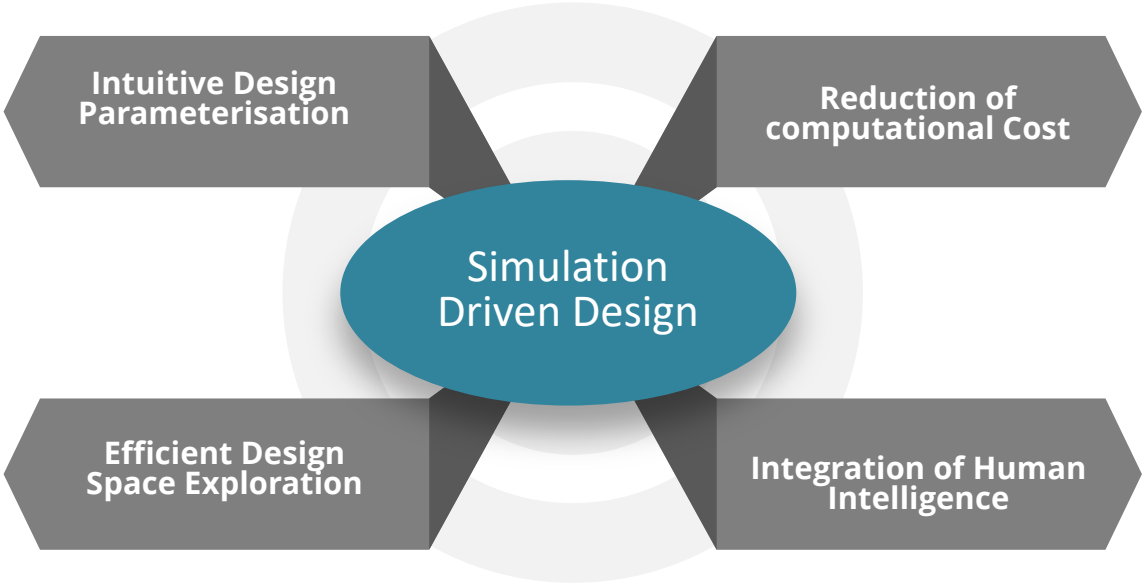


Figure 1.4: Requirements for effective simulation-driven design optimisation.

Chapter 2

Novelty

As depicted in Fig. 1.4, a compelling implementation of simulation-driven design optimisation necessitates a robust and intuitive design parameterisation for geometrically valid and feasible design variation. However, the design space resulting from this parameterisation should be versatile and low-dimensional while providing enhanced design diversity. However, the existing approaches used to create such spaces suffer from their intrinsic deficiencies, such as feature extraction-based techniques resulting in spaces that are prone to creating many infeasible designs and lack any notion of physics. In contrast, supervised approaches like PSA can overcome this, but their implementation has limitations, such as they suffer local instabilities and are computationally demanding. Furthermore, the exploration of design spaces should be more efficient and human-driven to facilitate designers and engineers throughout the product development to synthesise a diverse set of design ideas driven by engineers' performance requirements and designers' perceptions regarding designs' feasibility and form appearance.

To overcome the aforementioned challenges towards SDD, we make the following major contributions in this thesis (see the contribution summary in Fig. 2.1):

1. An *intra-sensitivity* technique to tackle the problem of local instabilities in PSA by identifying parameters whose perturbation has a significant impact on the sensitivity

index of the remaining parameters (chapter 4 [18]).

2. A *geometric moment-dependent sensitivity analysis* approach that harnesses the geometric variation of designs in a design space using geometric moments [14, 51] as a geometrical Quantity of Interest (QoI) to measure parametric sensitivities. This bypasses the need for evaluating expensive physical QoIs, and the results of the proposed approach can serve as a prior estimation of parametric sensitivities and use to construct a design space of lower dimension with only a subset of highly/strongly sensitive parameters for shape optimisation performed against physical QoI. (Chapter 5 [52]).

3. An FE-based *shape-supervised DSDR* approach which uses geometric moments to harness the compact geometric representation of the baseline shape and complement its physics during DSDR. Therefore, the resulting subspace has an enhanced capacity to produce a valid and diverse set of design alternatives, respectively, and is also physically informed to improve the convergence rate of the shape optimiser towards an optimal solution (Chapter 6 [53]).

4. A *feature-to-feature learning approach* to construct a physics-embedded lower-dimensional space for accelerating the training of surrogate models constructed with Gaussian process regression, which are later used to expedite the shape optimisation (Chapter 7 [21]).

5. A deep convolutional generative adversarial network, *ShipGAN*, to create a generic parametric modeller who can create geometrically valid and practically feasible alternatives of various ship hulls with the capability to transform one type of hull into a completely different one (Chapter 8).

6. A new interactive design system, *GenYacht*, which brings the benefit of the interactive and generative design to the preliminary design stage to generate user-driven hull forms with better performance (Chapter 9 [54]).

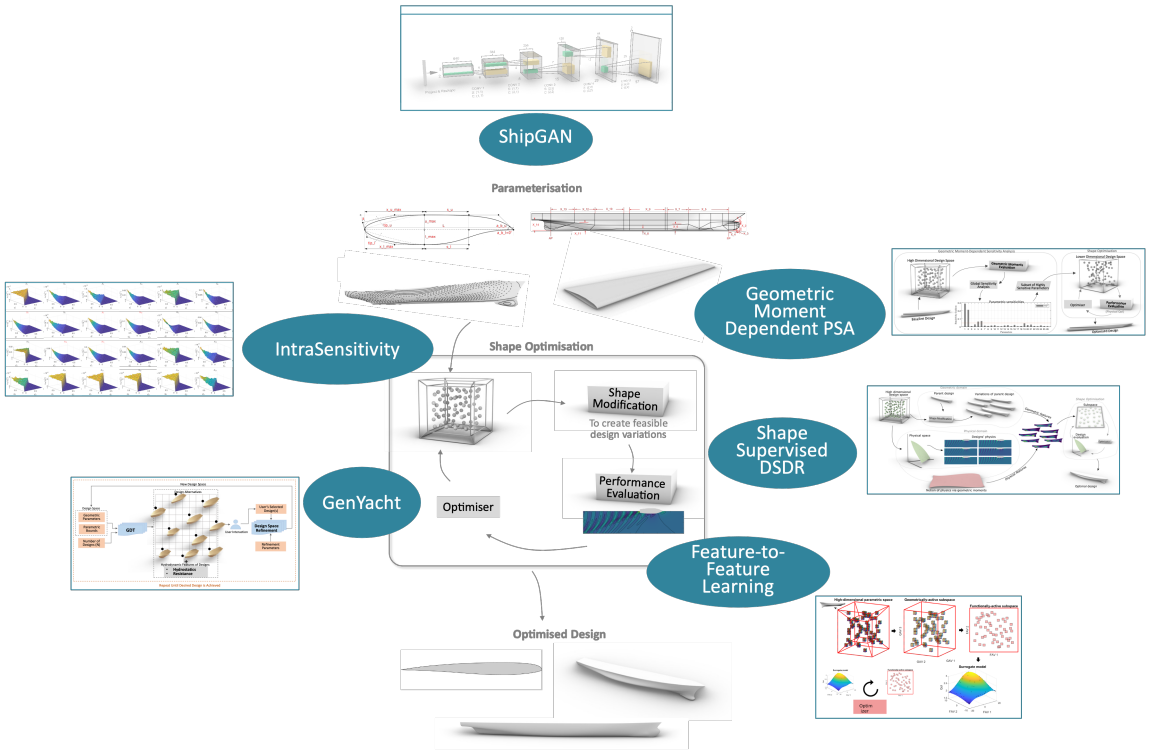


Figure 2.1: Contribution of thesis towards simulation-driven design optimisation.

2.1 Overview of proposed approaches

Intra-Senstivity. The concept of intra-sensitivity is introduced to identify parameters whose perturbation has a major impact on the sensitivity index of the remaining parameters. For this purpose, we first appeal to Active Subspace Method (ASM) and develop an ASM-based regional sensitivity analysis, which investigates parametric sensitivity in local regions of the design space and aids in extracting parameters' intra-sensitivity. This regional analysis is applied in conjunction with a Dynamic Propagation Sampling approach for tackling the computational complexity arising when high-dimensional problems are concerned. Once sensitive and intra-sensitive parameters are identified, then free-form features correlated to these parameters are evaluated using a feature saliency map built with the aid of Hausdorff distance.

Geometric moment-dependent PSA. To expedite PSA in the context of shape optimisation of free-form shapes. To leverage the computational burden that is likely to occur in engineering problems, we construct a shape signature vector and propose to use it as a substitute for physics. SSV is composed of shapes' integral properties, in our case geometric moments and their invariants of varying order (evaluated using the divergence theorem), and is used as quantity-of-interest (QoI) for prior estimation of parametric sensitivities. Opting for geometric moments is motivated by the fact that they are intrinsic properties of shapes' underlying geometry, and their evaluation is essential in many physical computations as they act as a medium for interoperability between geometry and physics.

Shape supervised DSDR. In shape optimisation problems, subspaces generated with conventional FE-based DSDR approaches often fail to extract the intrinsic geometric features of the shape that would allow the exploration of diverse but valid candidate solutions. More importantly, they also lack incorporation of any notion of physics against which shape is optimised. To simultaneously tackle these deficiencies, the proposed Shape supervised DSDR uses higher-level information about the shape in terms of its geometric integral properties, such as geometric moments and their invariants. Their usage is based on the

fact that moments of a shape are intrinsic features of its geometry, providing a unifying medium between geometry and physics. To enrich the subspace with latent features associated with the shape's geometrical features and physics, we also evaluate a set of composite geometric moments, using the divergence theorem, for appropriate shape decomposition. These moments are combined with the shape modification function to form a decomposed SSV uniquely representing a shape. Afterwards, the generalised Karhunen–Loève expansion is applied to SSV, embedded in a generalised (disjoint) Hilbert space, which results in a basis of the shape-supervised subspace retaining the highest geometric and physical variance.

Feature-to-feature learning. To release the computational burden of SDD, 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. In 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 terms of previously learned geometric features. Afterwards, geometric and functional features are augmented together to create a functionally-active subspace whose basis captures the geometric variance of designs and induces variability in the designs' physics. As the new subspace accumulates both the functional and geometric variance, it can be exploited for efficient design exploration and the construction of improved surrogate models for designs' physics prediction.

ShipGAN. In this work, we developed a ShipGAN model using deep convolutional generative adversarial networks (GANs) for the versatile parameterisation of ship hulls. For reliable training, we first select various categories of ship designs, including containers, oil tankers, bulk carriers, naval and crew supply vessels, etc., resulting in a shape dataset containing 52,591 designs. This strategy makes ShipGAN a generic parametric modeller with the ability to perform feasible and plausible design modifications for various hulls. The new model breaks the current conservatism in the parametric ship design paradigm,

where parametric modellers can only handle a particular ship type. We developed a new shape reconstruction strategy to convert all the training designs into a common geometric representation of the same resolution, as typical GANs can only take fixed dimensional vectors as input. A space-filling layer is placed right after the generator component to ensure that the trained generator can cover all design classes without a mode-collapsing issue. During training, designs are inputted in the form of a shape-signature tensor (SST), which harnesses the compact geometric representation using geometric moments and, for the ship design, induces the notion of physics. We have shown through extensive experimentation that ShipGAN can create designs with augmented features resulting in versatile design spaces that give geometrically valid and practically feasible shapes.

GenYacht. GenYacht, is proposed for creating optimal and user-centred yacht hull forms. GenYacht is a hybrid system involving generative and interactive design approaches, enabling users to create various design alternatives. Among them, a user can select a hull design with desirable characteristics based on its appearance and hydrostatics/hydrodynamic performance. GenYacht first explores a given design space using a generative design technique (GDT), which creates uniformly distributed designs satisfying the given design constraints. These designs are then presented to a user, and single or multiple designs are selected based on the user's requirements. Afterwards, based on the selections, the design space is refined using a novel space-shrinking technique (SST). In each interaction, SST shrinks the design space, which is then fed into GDT to create new designs in the shrank space for the next interaction. This shrinkage of design space guides the exploration process and focuses the computational efforts on user-preferred regions. The interactive and generative design steps are repeated until the user reaches a satisfactory design(s). The efficiency of GenYacht is demonstrated via experimental and user studies, and its performance is compared with interactive genetic algorithms.

2.2 Thesis outline

This thesis comprises six technical chapters discussing in depth the significant technical contributions of this thesis (see Fig. 2.2). Chapter three gives a background on current industry standards and their usage in SDD. In chapter 4, we explain our intra-sensitivity approach for identifying the local instabilities caused by the design parameterisation and design space during the implementation of PSA. Chapter 5 proposes a new geometric moment-dependent parametric sensitivity analysis approach that offloads the evaluation of parametric sensitivities from the physical quantity of interest (QoI) to relatively inexpensive geometrical QoI, which, compared to physical ones, are computationally less expensive to evaluate but provide essential clues about the form distribution and validity of the design, such as geometric moments and their invariants. A new FE-based DSDR approach, shape supervised DSDR, is proposed in Chapter 6, which uses a shape-signature vector composed of shape modification function and geometric moments, whose eigendecomposition provided a rich and physic-informed design lower-dimension subspace for accelerating the shape optimisation. In chapter 7, we work on approaches for expediting the construction of surrogate models with Gaussian process regression at an offline step. Chapter 8 explains our ShipGAN parametric model, which is based on a deep convolutional generative adversarial network. Finally, in Chapter 9, we present the GenYacht tool based on novel interactive and generative design techniques for user-centred and optimised exploration of yacht hull designs.

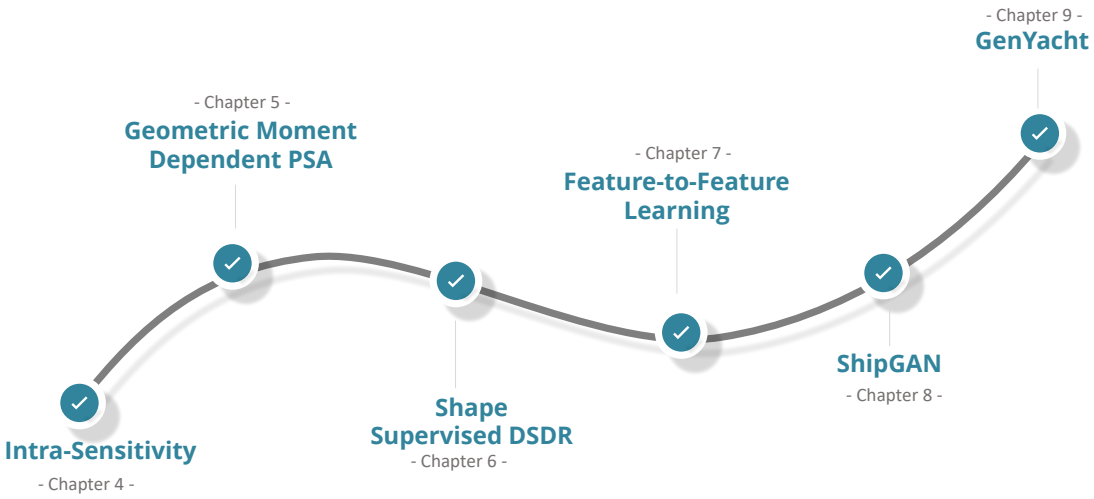


Figure 2.2: Thesis organisation.

Chapter 3

Background

3.1 Design parametrisation

Over the past decades, different application-specific and generic parametrisation techniques [55–57], rule-sets [58], and strategies [13, 35] have been proposed. Cagan et al. [57] described the parametrisation process as an iterative task, in which the decision on the selection of a suitable parametric approach is typically driven by the design's performance objective, its underline geometric representation, allowed number of design parameters and desired degree of design variability. For a typical engineering design, a *feature-based parametrisation* [55] is usually favoured, in which critical performance-associated features of the design are first identified and then parametrised with a rich set of parameters for *feasible* design variations [13, 59]. Compared to the free-form shapes, the parametrisation of solid models is generally straightforward, especially for those built from simple geometric primitives, such as mechanical components, and which result from a concurrent design process in which the designer creates models with respect to their manufacturing processes. For these models, identification of their features, such as holes, fillets, slots, chamfer, etc., is obvious, and the inherited parametrisation of these features, such as diameter of circle, radius of fillet, length and chamfer angle, etc., can create a complete parametric set [56]. These features are characterised by Langerak [34] as *regular form features* and are linked

with the manufacturing process. Such models may be parameterised in a free-form way, but their deformation may not always result in plausible design variations. A mechanical gear design can be an example of such components, typically constructed on feature-based approaches and are parameterised with common parameters such as the number of teeth, pitch circle diameter, face width, tooth thickness, etc., which are decided based on given design requirements. In contrast, features of a free-form shape can be theoretically unlimited [34], and their organic nature can be demanding for an efficient parametrisation.

In literature, different techniques, such as direct mesh-, basis vector-, domain element-, partial differential equation-, free-form deformation (FFD)-, polynomial and spline-based parametrisation, have been utilised for parametrisation of free-form shapes. A detailed description of these approaches can be found in [60,60]. Among these, FFD- [61] and spline-based techniques have been widely utilised in ship design. Some earlier and recent examples of these are [1, 4, 62–64]. Apart from these CAD-based technologies, an early attempt at parametric modelling of ships was made by Lackenby [65] in which hull variants are obtained by modifying the prismatic coefficient, the centre of buoyancy and the extent and position of the cylindrical mid-body of a hull. In spline-based approaches, control points defining the surface are often considered design parameters, which may ease the decision on the number of parameters, but the design modification with control points can be scruffy. The precise construction of features of free-form shapes often requires many patches, which have to join with adequate, at least G^1-, continuity. Such construction processes increase the number of control points. For instance, in the case of PD and FFD, the parent hull constructed with T-spline and NURBS is composed of 114 and 15,401 control points, respectively. Therefore, taking these control points as design parameters and performing design modifications based on them can introduce additional redundant control points, decrease smoothness and, more importantly, features of the modified shape may not retain their underlying structure. Therefore, these approaches are often augmented to create high-level parametrisation such as PD [4]. Although tightly application-dependent, PD couples free-form features with control points through linear procedural relations creating a low-

dimensional design space with a versatile set of design parameters. Some sophisticated examples of PD in ship design are [1, 4, 54].

It is noteworthy that the underlying principles used in PD and FFD techniques are different from each other. Therefore, shape optimisation performed with these techniques for the same design can lead to different results of varying quality [66]. Consequently, it is essential to study how PSA behaves when these two parametric approaches, in our case FFD and PD, are used. We are mainly interested to see which parametric technique is more efficient in detecting sensitive and intra-sensitive parameters.

3.1.1 Design space

Estimating parameters' sensitivity with respect to QoI mainly depends on the design space [30, 31]. Therefore, the identification of viable bounding values for parametric ranges is crucial. The resulting space should be robust enough to provide feasible design exploration and wide enough for design diversity to find the most optimal design [38]. For a typical parametric design problem, Krish [35] recommends forming the design space while keeping the initial design (in our case, the parent hull) at the centre of the space, representing this design as the most common one that can be generated. A ϵ-margin sampling-based approach was proposed by Chen, and Fuge [67], to set a feasible design space. It uses a data-driven probabilistic model to refine and expand the existing parametric ranges. Wu and Wang [68] proposed using knowledge-assisted models to set up an initial envelope of design space and guide the exploration towards its feasible regions. Khan and Awan [13] recommend that if no prior understanding of design specifications is available, a design space can be coarsely set up by assigning the lower and upper limits as a percentage of initial parametric values. Dogan et al., [69] in their generative design work, used a similarity metric, which set the design space based on the desired degree of diversity in the samples. The feasibility of the sampled designs was ensured with simple primitive constraints.

Following [35] and [13], this work design space is initially set up while keeping the baseline design at the centre and defining the lower and upper limit as a percentage of the

initial parameter values. It was further refined following Chen and Fuge's [67] strategy to minimise the possibility of invalid geometries. These invalid geometries are the sampled designs for which the parametric modeller or simulation tool may fail to provide results. These geometries can be excluded either by programming the internal constraints within the parametric modeller or by adjusting the bounding limits. A typical example of invalid free-form geometries is the designs with self-intersecting and non-watertight surfaces. Once an initial design space is formed, sensitivity and intra-sensitivity analyses are performed to identify the regions of designs contributing significantly towards the QoI and for the refinement of the parametric ranges.

3.2 Parametric sensitivity analysis

PSA has been widely used [70,71], improved [36], adopted [72], and hybridised [31,73] with other techniques for different purposes, but as stated in [74], one of its primary goals is to reduce the dimensionality of the design space by screening out the less sensitive parameters to designs' performance. It should be noted that there are two types of approaches in the field of parametric PSA, namely the local (LPSA) and global PSA (GPSA). LSA is usually derivative-based [31, 75], in which the change in QoI is evaluated against the variation of a single parameter. LSA investigates how a small perturbation near an input space value influences the value of QoI. On the contrary, in GSA, all parameters are varied simultaneously, and sensitivity is assessed over the entire range of each design parameter constructing the design space [37, 75]. In complex engineering problems, the influence of a design parameter may vary drastically as the remaining parameters change. Thus, it could be inadequate to evaluate the impact of a parameter on the QoI with the other parameters being kept constant. Accordingly, selecting significant parameters for shape optimisation based on their GSA is more appropriate in real-world applications [76] and used in the present work. The list of commonly used techniques is extensive, e.g., includes variance-based (or Sobol's method), derivative-based, density-based sensitivity and Morris

method. Interested readers can refer to [72, 77, 78] for a detailed overview and comparison of various PSA techniques.

3.2.1 Variance-based analysis

Among these techniques, variance-based methods like Sobol's analysis is suitable for complex nonlinear and non-additive models, therefore, are well received and utilised in different design applications. This method investigates how much of the overall variance of QoI is achieved due to the variability of a single or collection of design parameters. This variance is usually measured with *First-order indices* (or *main effects*) or *total-order indices* (or *total effects*). The former quantifies the direct contribution to QoI variance from an individual parameter, and the latter approximates the overall contribution from a parameter considering its immediate effect and interactions with all the other design parameters.

3.2.2 Active subspace method

An emerging advancement in reduced-order modelling and Global-PSA is ASM, which was proposed by Constantine [79, 80] and has been used for different applications, including ship hull design: see, e.g., [81–83]. ASM is based on the concept of discovering orthogonal active and inactive directions in the original design space for a certain QoI, where QoI have higher variability along with the active directions compared to the inactive ones. The active directions are used as the basis for constructing a new design space, referred to as *active subspace*, whose dimension is expected to be less than the original design space. Once discovered, active subspaces can be exploited to identify both local and global sensitivity of the problem's parameters on QoI with no additional computational cost [80]. As demonstrated in the present work, sampling-based estimation of sensitivity index with ASM is computationally less demanding and robust as it can provide stable results compared to Sobol's analysis, even with a small sample size.

3.2.3 Regional sensitivity analysis

Compared to PSA, regional sensitivity analysis is a new yet evolving field focusing on investigating the contribution of a parametric range to the variability of QoI. One of the earliest and well-structured regional analyses is the contribution to the Sample Mean Plot (CSM) proposed by Sinclair [84] and was further developed by Bolado-Lavin et al. [85]. The motivation behind CSM is that once the most important parameter is detected, the typical variance-based analyses do not facilitate the user in how any perturbation to the range of an input parameter influences the overall results. Therefore, CSM can be used to identify the local regions of the input space, which contributes highly towards the sample mean of QoI. Other recent extensions of CSM can be found in [86, 87].

Inspired by CSM, Tarantola et al. [36] proposed a Contribution to the Sample Variance plot (CSV). CSV are similar to the CSM; however, instead of mean, they use variance to infer the effect of local changes in design space on QoI variance. CSM and CSV evaluate the mean and variance over the different quantile ranges of a parameter and plot them against the cumulative distribution function of the parameter. If CSM or CSV are close to the diagonal, it indicates that mean or variance is constant over the entire range. If it shows fluctuation, then any perturbation in its range is more sensitive towards the mean or variance of QoI. Both CSM and CSV techniques facilitate users to quantify the variability of QoI over the range of a single parameter; however, it does not provide any information on how the sensitivity index of the parameter is varied if its range is changed and what impact it causes on the sensitivity of other parameters. On the contrary, the proposed regional analysis not only helps to identify the regions of design that account for the highest parametric sensitivity towards QoI but also identifies the parameters whose variation in range triggers the most decisive influence on the sensitivity of other parameters.

3.2.4 Computational complexity in PSA

The analytical implementation of GSA can often be tricky as it requires solving high-dimensional integrals. In this case, one has to appeal to sampling methods, such as Monte Carlo sampling (MCS) [37]. However, MCS is susceptible to slow convergence for stable results, as it requires evaluating a sufficiently large number of samples via computationally intensive physical simulations. Although a slightly better convergence rate can be obtained with quasi-Monte Carlo (Q-MC) or Latin-Hypercube (LH) methods, which are based on uniformly distributed design sequences, their advantage downgrade considerably in high-dimensional design spaces [37]. A sequential strategy, namely progressive Latin hypercube sampling (PLHS), was proposed by Sheikholeslami and Razavi [74]. As implied by its label, PLHS searches the design in the class of Latin hypercube and uses a criterion of space-filling to uniformly distribute the designs in a sub-set of sliced design spaces. Similar to [74], Wu [88] also utilised space-filling to propose a SA, which initially commences with samples obtained using the max-min principle of Latin hypercube, whose spread was improved by maximising a Euclidean distance with a coordination sorting algorithm. Gong et al. [89] compared different baseline sampling approaches, such as good lattice points, symmetric Latin hypercube uniformity, Ranked Gram-Schmidt and Quasi-Monte-Carlo, against uniformity scores and found out that a better convergence can be achieved with the first two approaches. Recently, Khan and Kaklis [18] proposed a Dynamic Sampling Strategy (DSS), which, along with space-filling, involves criteria of non-collapsing and repulsion. This method progressively increases the number of samples in each iteration; the non-collapsing maintains the diversity while repulsion helps create designs different from previously sampled ones in each iteration. DSS is proven to approximate a solution closer to the analytical one with a small sample size and, therefore, used in the present work.

As mentioned earlier, surrogate models such as non-parametric regression [90], polynomial chaos expansions [91], support vector machines [92], low-rank tensor approxima-

tions [93], Gaussian processes [16], and other Kriging methods [94] are also widely used for revealing parametric sensitivities with respect to costly physical quantities. Apart from the traditional surrogate modelling approaches, deep learning approaches [95] have recently gained attention specifically for quantifying key resource uncertainty in the system. Despite their proven efficiency, the sensitivity of parameters evaluated with these methods heavily depends on the accuracy of the surrogate models [16, 37].

Along with using retrofitted versions of the sampling methods to improve convergence and surrogate models to bypass expensive physics evaluation, few attempts have been made to ease the computational burden in the context of reducing the dimensions of a high-dimensional problem before performing PSA. For instance, Pronzato [73] used a Bayesian Linear Model constructed through a particular Karhunen–Loéve expansion to estimate Sobol's indices at a reduced computational cost. Furthermore, Sheikholeslami et al. [37] utilised a clustering-based strategy to ease the computational burden of implementing typical SA on high-dimensional design problems. Masood et al. [16] performed the eigendecomposition of the original design space using PCA and then proposed a method to drive the sensitivity of actual parameters from their lower-order projection. This method reduces the number of samples required to evaluate the robust sensitivity indices.

3.2.5 Instabilities in PSA

Apart from the high computational cost, another significant but often neglected challenge SA has to deal with is related to the fact that the sensitivity of parameters varies locally within the design space, meaning a parameter can be sensitive in some local regions of the design space but become insensitive in others. Such behaviour makes SA vulnerable to instabilities even with slight perturbation in the parametric ranges of the design space. Consequently, SA should be performed cautiously, especially at the preliminary design stage. At this stage, the design problem needs to be better established, and designers must be aware of the appropriate parametric ranges for performing a reliable sensitivity study. SA within a non-viable design space can be dismayed, eliminating essential parameters

from the design study.

To tackle this problem, we proposed intra-sensitivity to evaluate the behaviour of parametric sensitivity in local regions of the design space and to identify parameters whose perturbation in the range generates the most considerable inconsistency in the sensitivity of other parameters, respectively. Although these techniques can provide substantial aid to designers for reliable sensitivity studies, they are computationally demanding because of the extraction of the local behaviour of parametric sensitivities. Therefore, next, we appeal to a different direction to support the sensitivity study of design problems using quantities like geometric moment invariants, which, compared to physical criteria, are time inexpensive even for complex free-form or organic shapes but provide important clues regarding the physics. This work performs an extensive experimental study to prove that geometric moments can ease the designer in estimating parameters' sensitivity at the initial design stage.

3.3 Feature Extraction

Recently, GANs [32] have been used for reparameterisation of the shape with latent features, as baseline parameterisation approaches, like the ones used in free-form deformation, produce high-dimensional design spaces [18] that do not guarantee feasible/valid shapes. Chen et al. [32] proposed the Bézier-GAN for two-dimensional (2D) aerofoil design by introducing a Bézier layer into GAN to maximise subspace's representation capacity and compactness. However, the baseline parameterisation cannot automatically guarantee valid shapes and, as a result, these approaches require a training dataset of *existing designs*, which may prevent optimisers from finding innovative designs; a drawback studied in detail by Li and Zhang in [28]. Moreover, their usage can be problematic for novel problems, as, in this case, creating training datasets can be extremely arduous.

Furthermore, although subspaces resulting from the approaches described above may address the validity problem, physics-associated features still need to be present. The

Active Subspace Method [79] and supervised KLE [33] can handle this issue, but they become computationally intensive as they require direct evaluation of physics quantities and/or their gradients. Yonekura and Suzuki [26] recently used a conditional variational auto-encoder (CVAE) for aerofoil design. They used the lift coefficient as a condition to auto-encoder during training so that the decoder could generate the shape with specific performance. Chen and Ahmed [96] proposed PaDGAN to augment the design performance into the generator to create high-quality designs with good optimisation convergence. Another GANs-based method was proposed by Shu et al. [27], which elevates the quality of generated designs by iteratively updating the training dataset using performance-based design filtering. All methods above, i.e., [26, 27, 33, 79, 96] are efficient but supervised and therefore require performance labels to be evaluated for a large shapes dataset; if not readily available, the creation of such a dataset is computationally very demanding.

Chapter 4

Intra-sensitivity: Understanding local behaviour of parametric sensitivities

"*Limit risk with: Deep analysis Bargain purchase Sensitivity analysis.*"

Seth Klarman

4.1 Introduction

In this chapter, we develop and test a novel approach to tackle the difficulties inherent in implementing PSA, resulting in several key contributions. First, a *regional* extension of the *Active Subspace Method (ASM)* [79] (§4.2.2) is introduced and implemented to study how sensitivity indices tend to change when the range of design parameters, forming the design space, is modified. This helps in identifying local regions of the design space accounting for higher or lower parametric sensitivity towards QoI. Subsequently, the results obtained from the regional ASM are used to evaluate the proposed intra-sensitivity metric in order

to identify parameters whose perturbation in range causes the highest deviation on all sensitivity indices.

Furthermore, the impact of sensitive and intra-sensitive parameters on the design is quantified using a *feature saliency map* built with the aid of Hausdorff distance (§4.2.5). This provides a visual tool to identify the design features affected by these parameters and are characterised as sensitive or intra-sensitive features. Furthermore, a *Dynamic Propagation Sampling (DPS)* (§4.2.3) is utilised to circumvent the problems associated with sampling techniques, discussed in Chapter 3. DPS uses *space-filling* and *non-collapsing* criteria to sample as uniformly distributed designs as possible with few samples. Initially, sampling commences with a small set and iterations are performed to gradually increase the number of samples. The designs sampled in each iteration are constrained to be different from those sampled in the previous step using a new *repulsion* criterion.

To that end, a parent hull, which is similar to the KCS [1] container ship hull, is used for experiments; however, its length at the waterline (L_{wl}) is equal to 200m, whereas $L_{wl} = 232.5$m for the original KCS model. A pair of design spaces is generated with two different parametric approaches, namely *Free-Form Deformation (FFD)* [61] and *Procedural Deformation (PD)* [4]. FFD embeds an object within encompassing geometric primitives and modifies the object within these primitives as the surrounding lattice is modified. In contrast, PD builds an analytical relation of the design using high-level geometric parameters to create a feature-informed parametrisation. The sensitivity and intra-sensitivity of the parameters of the parent hull model parameterised with both techniques are evaluated and compared against the volume of displacement and the total resistance, which are critical criteria in ship design. Finally, the effectiveness of the approach is verified with different comparative studies. The main focus is given on the comparison of the convergence performance of the proposed regional analysis with existing regional and sampling techniques. The code to implement this approach is available at GitHub repository [2].

[1] http://www.simman2008.dk/KCS/container.html
[2] https://github.com/shahrozkhan66/IntraSens.git

The remainder of this chapter is organised as follows: Section 4.2 discusses the proposed approach to implement sensitivity and intra-sensitivity analyses. The numerical results of the proposed technique are given in Section 4.3. Concluding remarks and opportunities for future work are presented in Section 9.5.

4.2 Problem formulation

Let $\mathcal{G}_0$ be a geometric object representing a baseline design (e.g., a parent hull) in an ambient space $\mathcal{A} \subseteq \mathbb{R}^3$. We assume that $\mathcal{G}_0$ belongs to a rich class of objects in $\mathcal{A}$ parametrised with a set of n design parameters, $\mathbf{X} = \{X_k, k = 1, \ldots, n\} \subset \mathcal{X} \subseteq \mathbb{R}^n$, whose certain realisation is represented as $\mathbf{x} = \{x_k, k = 1, ..., n\}$ and is valued in appropriately defined intervals bound by lower ($\mathbf{x}_k^l$) and upper ($\mathbf{x}_k^u$) limits, $\mathbf{x}_k \in [\mathbf{x}_k^l, \mathbf{x}_k^u]$. These bounding limits form an n-dimensional solution/design space ($\mathcal{X}$) within which sensitivity analysis will be performed. Next, we assume that:

- We possess a parametric modeller $\mathcal{P}$ which, for any value $\mathbf{x} \in \mathcal{X}$, produces an object $\mathcal{G} = \mathcal{P}(\mathbf{x})$, and
- We possess a vector function $g : \mathcal{X} \subseteq \mathbb{R}^n \to \mathbb{R}$, which has continuous partial derivatives and is square integrable, or a simulation process that, for each $\mathbf{x} \in \mathcal{X}$, provides the output of interest.
- Finally, all the elements of $\mathbf{X}$ are also assumed to be statistically independent of each other, i.e., $p_{\mathbf{X}}(\mathbf{x}) = \prod_{k=1}^{n} p_{X_k}(x_k)$, where $p_{\mathbf{X}}(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}$ represents the Probability Density Function (PDF) of $\mathbf{X}$ and $p_{X_k}(x_k)$ is the marginal PDF of X_k.

4.2.1 Active subspace method

In this subsection, we discuss the theory and mathematical foundation of ASM.

Eigenspace: Let $\nabla_{\mathbf{x}} g$ to be the gradient of g for any realisation $\mathbf{x} \in \mathbb{R}^n$. That is,

$$g = g(\mathbf{x}), \quad \nabla_{\mathbf{x}} g = \nabla_{\mathbf{x}} g(\mathbf{x}) = \begin{bmatrix} \frac{\partial g}{\partial x_1}(\mathbf{x}) \\ \vdots \\ \frac{\partial g}{\partial x_n}(\mathbf{x}) \end{bmatrix}, \quad \text{for all } \mathbf{x} \in \mathcal{X}. \tag{4.1}$$

The objective here is to find an m-dimensional active subspace, where $m < n$. For this, the first step is to compute a covariance matrix $\mathcal{C}$ (given in Eq. (4.2)), which is a symmetric and positive semi-definite matrix and defined as an average of the outer product of $\nabla_{\mathbf{x}} g$ with itself:

$$\mathcal{C} = \int (\nabla_{\mathbf{x}} g)(\nabla_{\mathbf{x}} g)^T p_{\mathbf{X}}(\mathbf{x}) d\mathbf{x}. \tag{4.2}$$

Now, to identify the orthogonal active directions of $\mathcal{X}$, the eigenvectors of $\mathcal{C}$ are computed via its eigenvalue decomposition that can be written as

$$\mathcal{C} = \mathbf{W} \mathbf{\Lambda} \mathbf{W}^T, \tag{4.3}$$

where $\mathbf{W} = \{\mathbf{w}_k, k = 1, 2, \ldots, n\}$ is the $[n \times n]$ column matrix of orthogonal eigenvector ($\mathbf{w}_k \in \mathbb{R}^{1 \times n}$) with $\mathbf{w}_k^T \mathbf{w}_k = 1$, which defines the rotation of $\mathbb{R}^n$ and spans the basis of an eigenspace. Finally, $\mathbf{\Lambda} = \text{diag}(\lambda_k, k = 1, 2, \ldots, n)$ are the eigenvalues sorted in descending order $\lambda_1 \geq \lambda_2 \geq, \ldots, \lambda_n \geq 0$.

Active and inactive subspaces: The eigendecomposition of $\mathcal{C}$ reveals the following structure on g ([79], Lemma 3.1)

$$\lambda_k = \int_{\mathcal{X}} \left(\mathbf{w}_k^T \nabla_{\mathbf{x}} g(\mathbf{x}) \right)^2 p_{\mathbf{X}}(\mathbf{x}) d\mathbf{x}, \quad k = 1, \ldots, n, \tag{4.4}$$

which represents that the k^{th} eigenvalue, $\lambda_k \in \mathbb{R}$, measures the average squared directional derivative of g along its corresponding eigenvector, $\mathbf{w}_k$. This means if $\lambda_k > \lambda_{k+1}$ then on average the mean-squared change in g upon perturbing $\mathbf{X}$ along $\mathbf{w}_k$ is higher than $\mathbf{w}_{k+1}$. As g is continuous, so if λ_{k+1} is equal to zero then the directional derivative, $\nabla_{\mathbf{x}} g^T \mathbf{w}_{k+1}$,

is zero. In other words, g is constant/flat along the direction defined by the $\mathbf{w}_{k+1}$, which can be ignored for dimension reduction. Thus, to form a reduced order basis, the sorted $\mathbf{W}$ and $\mathbf{\Lambda}$ are partitioned into two sets, containing active and inactive directions as

$$\mathbf{\Lambda} = \begin{bmatrix} \mathbf{\Lambda}_1 & \\ & \mathbf{\Lambda}_2 \end{bmatrix}, \quad \mathbf{W} = \begin{bmatrix} \mathbf{W}_1 & \mathbf{W}_2 \end{bmatrix}. \tag{4.5}$$

The columns of $\mathbf{W}_1 = \{\mathbf{w}_i, i = 1, 2, \ldots, m\}$ and $\mathbf{W}_2 = \{\mathbf{w}_j, j = 1, 2, \ldots, n-m\}$ are the dominant and non-dominant elements of $\mathcal{C}$, which span the basis of the active and inactive design spaces, receptively, and $\mathbf{\Lambda}_1$ and $\mathbf{\Lambda}_2$ are their eigenvalues. Here, $\mathbf{W}_1$ contains first m columns of $\mathbf{W}$ corresponding to the first largest m eigenvalues, $\mathbf{\Lambda}_1 = \{\lambda_i, i = 1, 2, \ldots m\}$. Once identified, $\mathbf{x}$ can be projected on these subspaces using Eq. (4.6) to find its active $\mathbf{y} = \{y_i, i = 1, 2, \ldots, m\}$ and inactive $\mathbf{z} = \{z_j, j = 1, 2, \ldots, n-m\}$ parameters:

$$\mathbf{y} = \mathbf{W}_1^T \mathbf{x} \in \mathbb{R}^m, \quad \mathbf{z} = \mathbf{W}_2^T \mathbf{x} \in \mathbb{R}^{n-m}. \tag{4.6}$$

Among $\mathbf{y}$ and $\mathbf{z}$, we are only interested in $\mathbf{y}$ as its basis $\mathbf{W}_1$ covers the largest variability of g and it is negligibly influenced by inactive variable, $\mathbf{z}$.

Sensitivity: After identifying the active subspace, the global sensitivity of each design parameter in $\mathbf{X} = \{X_k, k = 1, 2, \ldots, n\}$ can be derived using the sensitivity indices denoted as $\alpha(m)$. The sensitivity index, $\alpha_k(m)$, of the k^{th} design parameter is calculated as

$$\alpha_k = \alpha_k(m) = \sum_{i=1}^{m} \lambda_i w_{k,i}^2, \quad \forall k \in \{1, 2, \ldots, n\}. \tag{4.7}$$

The sensitivity indices in Eq. (4.7) help to measure the contribution of design parameters towards the variability of output of g. A parameter with a high sensitivity index is more sensitive/significant than a low one. Once identified, the insignificant parameters (i.e., parameters with a low sensitivity index) can be fixed to reduce dimensionality. Furthermore, the range of significant parameters can be refined to reduce uncertainty in the output of

g. To visually compare indices, α_k is normalised as

$$\alpha_k \mapsto \frac{|\alpha_k|}{\sqrt{\sum_{k=1}^{n}(\alpha_k)^2}}, \tag{4.8}$$

which removes the signs and scales the vector to have norm one. It is noteworthy that *when sensitivity indices are learnt on eigenspace* $\mathbf{W}$ *instead of* $\mathbf{W}_1$ *then it is similar to the local sensitivity analysis*. This is an important feature of ASM that one can evaluate both local and global indices with no additional computational burden.

4.2.2 Quantifying Intra-Sensitivity

This subsection gives details of mathematical formulation and general assumptions set for the proposed regional ASM.

Proposed regional ASM: A regional version of ASM can be formulated to study how a change in the range of a parameter affects the overall sensitivity of other parameters. Later, these results can be used to assess the intra-sensitivity index of the parameters. To commence, the covariance matrix in Eq. (4.2) can be rewritten in more detail as

$$\mathcal{C} = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} (\nabla_{\mathbf{x}} g)(\nabla_{\mathbf{x}} g)^T \prod_{k=1}^{n} p_{X_k}(x_k) dx_k. \tag{4.9}$$

For regional analysis, the above matrix is evaluated over a reduced range of the parameters. Suppose the range of parameter, X_k, is reduced from $[-\infty, \infty]$ to $\left[-\infty, F_k^{-1}(q_k^u)\right]$ or to $\left[F_k^{-1}(q_k^l), \infty\right]$, where $F_k^{-1}(\cdot)$ is the inverse cumulative distribution of X_k at quantiles q_k^l and q_k^u, with $q_,^l q_k^u \in [0,1]$. Then the PDF of X_k is updated as follows:

$$p_{X_k}^u(x_k) = \begin{cases} \frac{p_{x_k}(x_k)}{\int_{-\infty}^{F_k^{-1}(q_k^u)} p_{x_k}(x) dx} & \text{if } \; x_k \in \left[-\infty, F_k^{-1}(q_k^u)\right], \\ 0 & \text{elsewhere,} \end{cases} \tag{4.10a}$$

$$p^l_{X_k}(x_k) = \begin{cases} \frac{p_{x_k}(x_k)}{\int_{F_k^{-1}(q_k^l)}^{\infty} p_{x_k}(x)dx} & \text{if } \; x_k \in \left[F_i^{-1}(q_k^l), \infty\right], \\ 0 & \text{elsewhere.} \end{cases} \tag{4.10b}$$

Under these new settings, the corresponding covariance matrices can be computed as

$$\mathcal{C}_k(q_k^u) = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} \int_{-\infty}^{F_k^{-1}(q_k^u)} (\nabla_{\mathbf{x}} g)(\nabla_{\mathbf{x}} g)^T p^u_{X_k}(x_k) dx_k \prod_{r=1, r\neq k}^{n} p_{X_r}(x_r) dx_r, \tag{4.11a}$$

$$\mathcal{C}_k(q_k^l) = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} \int_{F_k^{-1}(q_k^l)}^{\infty} (\nabla_{\mathbf{x}} g)(\nabla_{\mathbf{x}} g)^T p^l_{X_k}(x_k) dx_k \prod_{r=1, r\neq k}^{n} p_{X_r}(x_r) dx_r. \tag{4.11b}$$

In Eq. (4.11), multiple integrals are evaluated over the entire range, $[-\infty, \infty]$, for all the parameters expect X_k, for which the integral is evaluated either over $\left[-\infty, F_k^{-1}(q_k^u)\right]$ or $\left[F_k^{-1}(q_k^l), \infty\right]$. In similarity to Eq. (4.3), the eigendecomposition of these covariance matrices is evaluated as

$$\mathcal{C}_k(q_k^u) = \mathbf{W}_k^u \mathbf{\Lambda}_k^u \mathbf{W}_k^{uT}, \quad \mathcal{C}_k(q_i^l) = \mathbf{W}_k^l \mathbf{\Lambda}_k^l {\mathbf{W}_k^l}^T, \tag{4.12}$$

where, $\mathbf{W}_k^u$, $\mathbf{\Lambda}_k^u$ and $\mathbf{W}_k^l$, $\mathbf{\Lambda}_k^l$ denote the eigenvector and eigenvalue matrices evaluated over the reduced range, $\left[-\infty, F_k^{-1}(q_k^u)\right]$ and $\left[F_k^{-1}(q_k^l), \infty\right]$, of X_k, respectively. As a result of this decomposition, the corresponding sensitivity indices of all parameters can be expressed as:

$$\alpha_{k'}(q_k^u) = \sum_{i=1}^{m} \lambda_i^u \; (w^u_{k',i})^2, \quad \alpha_{k'}(q_k^l) = \sum_{i=1}^{m} \lambda_i^l \; (w^l_{k',i})^2, \quad \forall k' \in \{1, 2, \ldots, n\}. \tag{4.13}$$

If $k' = k$ then $\alpha_{k'}(q_k^u)$ and $\alpha_{k'}(q_k^l)$ represent the sensitivity index calculated over the reduced range of X_k, while if $k' \neq k$ then $\alpha_{k'}(q_k^u)$ and $\alpha_{k'}(q_k^l)$ represent the sensitivity indices of the remaining parameters in $\mathbf{X}$ calculated over the reduced range of X_k. After evaluating

$\alpha_k(q_k^u)$ and $\alpha_k(q_k^l)$, $q_k^u, q_k^l \in [0,1]$, one can plot them on a 2D space, $[0,1]^2$, with q_k^u and q_k^l on x-axis representing a fraction of distribution range of X_k. If the computational cost permits, a regional analysis can be performed while simultaneously varying q_k^u and q_k^l. In this case, the range of X_k is reduced from $[-\infty, \infty]$ to $\left[F_k^{-1}(q_k^l), F_k^{-1}(q_k^u)\right]$ and along with the constraint $q_k^l < q_k^u$, a three-dimensional (3D) plot can be created in $[0,1]^3$ with q_k^l/q_k^u on the x-/y-axis, respectively, and $\alpha_k(q_k^{l,u})$ on the z-axis. Then the covariance matrix can be formulated as:

$$\mathcal{C}_k(q_k^l, q_k^u) = \mathcal{C}_k(q_k^u) - \mathcal{C}_k(q_k^l) = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} \int_{F_k^{-1}(q_k^l)}^{F_k^{-1}(q_k^u)} (\nabla_{\mathbf{x}} g)(\nabla_{\mathbf{x}} g)^T p_{X_k}^*(x_k) dx_k \prod_{r=1, r\neq k}^{n} p_{X_r}(x_r) dx_r, \quad (4.14)$$

where the PDF of X_k is evaluated as:

$$p_{X_k}^*(x_k) = \begin{cases} \frac{p_{x_k}(x_k)}{\int_{F_k^{-1}(q_k^l)}^{F_k^{-1}(q_k^u)} p_{x_k}(x) dx} & \text{if } \; x_k \in \left[F_k^{-1}(q_k^l), F_k^{-1}(q_k^u)\right] \\ 0 & \text{elsewhere,} \end{cases} \quad (4.15)$$

and the corresponding sensitivity expressed as:

$$\alpha_{k'}(q_k^l, q_k^u) = \alpha_{k'}(q_k^u) - \alpha_{k'}(q_k^l) = \sum_{i=1}^{m} \lambda_i^{l,u} \; (w_{k',i}^{l,u})^2 \quad \forall k' \in \{1, 2, \ldots, n\}, \quad (4.16)$$

where $\lambda_i^{l,u}$ and $w_{k',i}^{l,u}$ are the elements of the matrices $\mathbf{\Lambda}_k^{l,u}$ and $\mathbf{W}_k^{l,u}$, respectively, obtained from the eigendecomposition of $\mathcal{C}_k(q_k^l, q_k^u)$.

Both 2D and 3D regional plots, which will be discussed in § 4.3.2, can help designers to analyse the behaviour of parameters' sensitivity indices over their entire variability range. Furthermore, these analyses also facilitate the identification of the regions where a certain

parameter has the least and maximum impact towards the variability of output of g. With these data in hand, designers can make informed decisions on how to set the bounding limits of design parameters while having the highest or reduced impact of certain parameters on the overall performance.

Intra-sensitivity: As mentioned earlier, the reduction in the range of one parameter may affect not only its sensitivity index but also the sensitivity indices of other parameters. The study of the effect of any parameter's range on the sensitivity of remaining parameters can itself be a sensitivity study. This could identify the parameter whose reduction in the range may cause the highest deviation in the sensitivity index of the entire set of parameters, and this is what we call intra-sensitivity. More specifically, for the parameter X_k the intra-sensitivity index, I_k, can be measured as

$$I_k = \sum_{k'=1,k'\neq k}^{n} ||\alpha_{k'}^{max}(q_k^l, q_k^u) - \alpha_{k'}^{min}(q_k^l, q_k^u)||, \quad \forall k \in \{1, 2, \ldots, n\}, \tag{4.17a}$$

$$\alpha_{k'}^{max}(q_k^l, q_k^u) = max\left(\alpha_{k'}(q_k^l, q_k^u),\ \forall q_k^l, q_k^u \in [0,1] \mid q_k^l < q_k^u\right), \quad \forall k' \in 1, 2, \ldots, n \tag{4.17b}$$

$$\alpha_{k'}^{min}(q_k^l, q_k^u) = min\left(\alpha_{k'}(q_k^l, q_k^u),\ \forall q_k^l, q_k^u \in [0,1] \mid q_k^l < q_k^u\right), \quad \forall k' \in 1, 2, \ldots, n \tag{4.17c}$$

For simplicity, the expressions, $\alpha_{k'}(q_k^l, q_k^u)$, $\alpha_{k'}^{max}(q_k^l, q_k^u)$ and $\alpha_{k'}^{min}(q_k^l, q_k^u)$, are represented as $\alpha_{k'}(q_k^{l,u})$, $\alpha_{k'}^{min}(q_k^{l,u})$ and $\alpha_{k'}^{max}(q_k^{l,u})$, respectively. To differentiate, the sensitivity index can be defined as *a measure of the impact of the variability of an input parameter on the variability of QoI*. In contrast, the intra-sensitivity index is *a measure of the impact of the variability of an input parameter on the sensitivity index of other input parameters*.

4.2.3 Tackling the computational complexity

The elements of the covariance matrix, $\mathcal{C}$, are $n-$dimensional integrals, so its eigendecomposition for evaluation of $\mathbf{W}$ and $\mathbf{\Lambda}$ require evaluating high-dimensional integrals, which can be difficult, if not impossible, to evaluate. Although one could use deterministic numerical integration methods, they are unsuitable for high-dimensional problems, especially

if evaluating g is computationally costly [79, 97]. Therefore, eigenpairs of $\mathcal{C}$ are approximated using Eq. (4.18) with sampling methods, where samples are drawn from density, $p_{\mathbf{X}}(\mathbf{x})$ [79].

$$\mathcal{C} \approx \frac{1}{N} \sum_{b=1}^{N} \nabla_{\mathbf{x}} g(\mathbf{x}_b) \ \nabla_{\mathbf{x}} g(\mathbf{x}_b)^T \tag{4.18}$$

Here, N is the number of design instances sampled from $\mathcal{X}$, which, along with their performance output (Y), create a training dataset, $\mathcal{D}$, and $\nabla_{\mathbf{x}} g(\mathbf{x}_b)$ is the gradient vector (see Eq. (4.1)) for b^{th} design, $\mathbf{x}_b$, in $\mathcal{D}$. The approximation accuracy of $\mathcal{C}$ depends on the distribution of N design over $\mathcal{X}$ [80]. Additionally, evaluation of $\mathcal{C}$ requires the gradients, $\nabla_{\mathbf{x}} g$, which can be estimated with various techniques in the literature. For instance, if numerical noise in g is small enough and if the baseline legacy code or simulation allows, one could use finite-difference or adjoint solvers. However, often these capabilities are absent when simulation includes multiple or coupled components. Moreover, for a complex problem involving free-form shapes like ships, $\nabla_{\mathbf{x}} g$, especially evaluated from adjoint solvers, tends to be qualitatively correct (i.e., they have right relative scaling and right sign). However, quantitatively, in terms of magnitude, they are not reliable [97, 98], at least when they are used for ASM.

For discovering the active subspaces, we are interested in the local behaviour of the problem; therefore, $\nabla_{\mathbf{x}} g$ can be approximated with a local surrogate model. Inventors of ASM proposed a heuristic approach for building *local model-based* gradients, in which a local linear model is fitted within a subset of predictions from the training dataset [79]. In this approach, to evaluate the gradients for b^{th} design, $\mathbf{x}_b$, we obtain a subset of ℓ designs from $\mathcal{D}$ nearest to $\mathbf{x}_b$. A local model is then fitted on this subset and its gradients are evaluated. This process is repeated for all N samples in $\mathcal{D}$.

Therefore, the appropriate selection of N is crucial for the sensitivity indices' robustness and stable design parameters' ranking. On the one hand, small N will not accurately estimate the eigenpairs; thus, it will result in an unstable ranking. On the other hand,

large N will impose a computational barrier as it will require running a large number of computationally demanding physical simulations. Even though one could use a heuristic metric that estimates N needed to approximate the spectrum of sums of random metrics, such as $N = \gamma n \log(n)$, where $\gamma \in [1, 10]$ [79]. But this could still spawn computationally demanding sensitivity evaluation. Therefore, we use an iterative approach that progressively increases N and tracks the ranking of the parameters as described in Algorithm 1. In this approach, the sampling is started with $N = \gamma n \log(n)$ samples, and sensitivity ranking is obtained. At each iteration, new samples are added to $\mathcal{D}$ and iterations are terminated when we reach the maximum number of allowable samples (N_{allow}) or sensitivity ranking remains invariant for five consecutive iterations. It should be stressed that if the latter criterion is used, the sensitivity ranking of two insensitive parameters with small or similar sensitivity indices may deviate even with a slight change in their sensitivity index. Therefore, instead of terminating the Algorithm 1 with the ranking of all parameters, the termination criterion involves only the significant parameters.

It should be noted that convergence of Algorithm 1 may slow down if MC-based sampling is used because of the aforementioned reasons in §3.2. Furthermore, as shown in the subsequent sections, sampling similar designs in the subsequent iterations may prolong convergence as no new information is added to expedite it. Therefore, we adopted Dynamic Propagation Sampling (DPS), which is based on Khan and Gunpinar's technique [38] and samples uniformly distributed and diverse samples at each iteration.

Dynamic propagation sampling

Let the design space $\mathcal{X}$ be bounded by the lower $\mathbf{x^l}$ and upper $\mathbf{x^u}$ bounds of design parameters (i.e., $\mathcal{X} := \{x_k^l \leq x_k \leq x_k^u, \forall k \in \{1, 2, \dots n\}\}$). During sampling, our objective is to explore $\mathcal{X}$ in order to find a set $\mathcal{S}$ consisting of N samples ($\mathcal{S} = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \dots, \mathbf{x}_N\} \in \mathcal{X}$) while incorporating the criteria of space-filling, non-collapsing and repulsion.

Space-filling criterion: This criterion ($F_1(\mathcal{S})$) is implemented using Audze and Eglais [99] approach, which follows a physical analogy that molecules, designs in our case, in

Algorithm 1 The pseudo-code of sensitive ranking algorithm

1: **function** *sensitivity*$(\mathcal{X}, \mathcal{G}_0, \mathcal{P}, g)$
2: **Input:** Create a baseline design, $\mathcal{G}_0$, and parametrise it with n parameters to create a parametric modeller, $\mathcal{P}$.
3: **Input:** Select a suitable function or simulation model, g, to evaluate QoI.
4: **Input:** Define the design space with lower and upper bounds of n parameters, $\mathcal{X} := \{x_k^l \leq x_k \leq x_k^u,\ \forall k \in \{1, 2, \dots n\}\}$.
5: Initialise $\nabla\mathcal{D}_{total} \leftarrow \emptyset$ to store the gradients over iterations, $N \leftarrow \text{round}(\gamma\ n\ \log(n))$, iteration count $(i \leftarrow 0)$ and $\hat{\mathcal{S}} \leftarrow \emptyset$ to store samples of each iteration.
6: Set N_{allow} based on the allowable computational budget.
7: **while** $size(\nabla\mathcal{D}_{total}) \leq N_{allow}$ **do**
8: $i \leftarrow i + 1$
9: Generate sampling set $\mathcal{S} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N]^T$ using Algorithm 2.
10: Evaluate $g(\mathbf{x})$ for all elements of $\mathcal{S}$ and a create set, $\mathcal{Y} = [Y_1, Y_2, \dots, Y_N]^T$, containing its outputs.
11: Create dataset $\mathcal{D} = [\mathcal{S}\ \ \mathcal{Y}]$ consisting of columns of $\mathcal{S}$ as independent variables and $\mathcal{Y}$ as dependent variable.
12: Evaluate set $\nabla\mathcal{D} = [\nabla_{\mathbf{x}} g(\mathbf{x}_1), \nabla_{\mathbf{x}} g(\mathbf{x}_2), \dots, \nabla_{\mathbf{x}} g(\mathbf{x}_N)]^T$, which contains gradients for all N elements of $\mathcal{D}$.
13: $\nabla\mathcal{D}_{total} \leftarrow \nabla\mathcal{D}_{total} \cup \nabla\mathcal{D}$
14: Compute $\mathcal{C}$ with elements of $\nabla\mathcal{D}_{total}$ and its eigenvalue decomposition as $\mathcal{C} \approx \frac{1}{N}\sum_{b=0}^{N} \nabla_{\mathbf{x}} f(\mathbf{x}_b)\ \nabla_{\mathbf{x}} f(\mathbf{x}_b)^T = \mathbf{W}\mathbf{\Lambda}\mathbf{W}^T$
15: Partition eigenspace of $\mathcal{C}$: $\mathbf{\Lambda} = \begin{bmatrix} \mathbf{\Lambda}_1 & \\ & \mathbf{\Lambda}_2 \end{bmatrix}$, $\mathbf{W} = \begin{bmatrix} \mathbf{W}_1 & \mathbf{W}_2 \end{bmatrix}$
16: Calculate sensitivity index set, $\boldsymbol{\alpha}^i$, at i^{th} iteration for all parameters, $\boldsymbol{\alpha}^i = \{\alpha_k, \forall k = 1, 2, \dots, n\}$.
17: Obtain ranking, $\mathbf{R}^i$, of $\mathbf{X} = \{X_k, k = 1, 2, \dots, n\}$ at the i^{th} iteration based on $\boldsymbol{\alpha}^i$
18: Store $\mathbf{R}^i$, $\boldsymbol{\mathcal{R}} \leftarrow \mathbf{R}^i$
19: **if** $i \geq 5$ **then**
20: **if** last five elements of $\boldsymbol{\mathcal{R}}$, $\{\mathbf{R}^{i-4}, \mathbf{R}^{i-3}, \dots, \mathbf{R}^i\}$, are identical **then**
21: **return** $\boldsymbol{\alpha}^i$ and $\mathbf{R}^i$
22: **end if**
23: **end if**
24: **if** $i = 1$ **then**
25: $N \leftarrow N/2$.
26: **end if**
27: $\hat{\mathcal{S}} \leftarrow \hat{\mathcal{S}} \cup \mathcal{S}$ {//Samples $\hat{\mathcal{S}}$ will be used in next iteration at step 9 to sample $\mathcal{S}$ different from $\hat{\mathcal{S}}$.}
28: **end while**
29: **return** $\boldsymbol{\alpha}^i$ and $\mathbf{R}^i$

space exert repulsive forces on each other that lead to potential energy in a space. These molecules are in equilibrium in case of minimum potential energy, which guarantees their uniform distribution over the entire space. This criterion for N design is evaluated as.

$$F_1(\mathcal{S}) = \sum_{p=1}^{N-1} \sum_{q=p+1}^{N} \frac{1}{\mathcal{M}(\mathbf{x}_p, \mathbf{x}_q)^2}, \tag{4.19a}$$

$$\mathcal{M}(\mathbf{x}_p, \mathbf{x}_q) = \sqrt{\sum_{k=1}^{n} (x_{p,k} - x_{q,k})^2}. \tag{4.19b}$$

Here, $\mathcal{M}(\mathbf{x}_p, \mathbf{x}_q)$ is the Euclidean norm between the pair of designs, p and q. Minimisation of $F_1(\mathcal{S})$ favours their uniform distribution in $\mathcal{X}$.

Non-collapsing criterion: In the case of high-dimensional design spaces, the space-filling criterion favours the placement of designs towards the boundaries of the design space. Therefore, along with space-filling, a criterion of non-collapsing is also incorporated during sampling to ensure an even proportion of samples over $\mathcal{X}$. This criterion divides the range of each element of $\mathbf{X}$ into N intervals and constraints the placement of more than one design in the same interval. It is incorporated into the search process using Eq. (4.20), which calculates the number of intervals that N designs share. Minimising this equation can lead to either complete or quasi-non-collapsing designs depending on a user-controlled parameter ω, which adjusts the relative weight of $F_2(\mathcal{S})$.

$$F_2(\mathcal{S}) = \omega \sum_{p=1}^{N-1} \sum_{q=p+1}^{N} \mathcal{K}(\mathbf{y}_p, \mathbf{y}_q), \tag{4.20a}$$

$$\mathcal{K}(\mathbf{y}_p, \mathbf{y}_q) = \sum_{j=1}^{n} f(y_{p,k}, y_{q,k}), \tag{4.20b}$$

$$f(y_{p,k}, y_{q,k}) = \begin{cases} 1 & \text{if } y_{p,k} = y_{q,k} \\ 0 & \text{otherwise} \end{cases}. \tag{4.20c}$$

In Eq. (4.20), $\mathcal{K}(\mathbf{y}_p, \mathbf{y}_q)$ denotes the number of intervals that designs p and q share, and $\mathbf{y}_p$ and $\mathbf{y}_q$ are the discrete position for $\mathbf{x}_p$ and $\mathbf{x}_q$, respectively. For each k^{th} design parameter, $x_{p,k}$, of p^{th} design its range between the upper and lower limits is partitioned in N intervals and $y_{p,k}$ is the order of sub-interval that contains $x_{p,k}$.

Repulsion criterion: To maintain diversity in the dataset, designs sampled in each iteration of Algorithm 1 should be different than the previously sampled ones; otherwise, no new information will be produced, which may result in unnecessary computational cost. Therefore, along with space-filling and collapsing, a repulsion criterion is introduced to sample the previously unexplored spaces of the design space, which is implemented as

$$F_3(\mathcal{S}, \hat{\mathcal{S}}) = \sum_{p=1}^{N} \sum_{q=1}^{\hat{N}} \frac{1}{\hat{\mathcal{M}}(\mathbf{x}_p, \hat{\mathbf{x}}_q)^2}, \tag{4.21a}$$

$$\hat{\mathcal{M}}(\mathbf{x}_p, \hat{\mathbf{x}}_q) = \sqrt{\sum_{k=1}^{n} (x_{p,k} - \hat{x}_{q,k})^2}, \tag{4.21b}$$

and follows an analogy of a typical repulsion process in physics. In which minimising $F_3(\mathcal{S}, \hat{\mathcal{S}})$ increases repulsive forces, under the influence of which designs in $\mathcal{S}$ tend to move away from designs of $\hat{\mathcal{S}}$. Similar to Eq. (4.19b), $\hat{\mathcal{M}}(\mathbf{x}_p, \hat{\mathbf{x}}_q)$ is the Euclidean norm between p^{th} newly sampled design, $\mathbf{x}_p$, and q^{th} previously sampled design, $\hat{\mathbf{x}}_q$. $\hat{N}$ is the total number of previously sampled designs in set $\hat{\mathcal{S}}$.

Minimising $F_3(\mathcal{S}, \hat{\mathcal{S}})$ makes the newly sampled design to be apart from the previously sampled designs, and $F_1(\mathcal{S})$ tries to distribute these designs over the hitherto unexplored regions uniformly. Considering the three criteria introduced above, we set up a minimisation problem for the objective function $\mathbf{F}(\mathcal{S}, \hat{\mathcal{S}})$ as

$$\min_{\mathcal{S}} \quad \mathbf{F}(\mathcal{S}, \hat{\mathcal{S}}) = \sum_{p=1}^{N-1} \sum_{q=p+1}^{N} \frac{1}{\mathcal{M}(\mathbf{x}_p, \mathbf{x}_q)^2} \quad + \quad \omega \sum_{p=1}^{N} \sum_{q=p+1}^{N} \mathcal{K}(\mathbf{y}_p, \mathbf{y}_q) + \sum_{p=1}^{N} \sum_{q=1}^{\hat{N}} \frac{1}{\hat{\mathcal{M}}(\mathbf{x}_p, \hat{\mathbf{x}}_q)^2} \tag{4.22}$$

$$\text{subject to} \quad \mathcal{S} \in \mathcal{X}$$

The above optimisation problem is solved with Khan and Gunpinar's [38] metaheuristic-based approach. It commences with an initial population, $\mathbf{P}$, consisting of N sub-populations, $\mathbf{P} = \{p_L, L = 1, 2, \ldots N\}$. The L^{th} sub-population, p_L, of $\mathbf{P}$ consists of n_s randomly sampled designs as $p_L = \{\mathbf{x}_c, c = 1, 2, \ldots n_s\}$. The position of these designs is then optimised following the stepwise procedure mentioned in Algorithm 2. The termination criterion for optimisation can be the maximum number of function evaluations or allowable optimisation iterations. At the end of the optimisation, a sampling run will be completed with an optimal sample set $\mathcal{S}$ of N space-filling, non-collapsing and repulsive designs at the output.

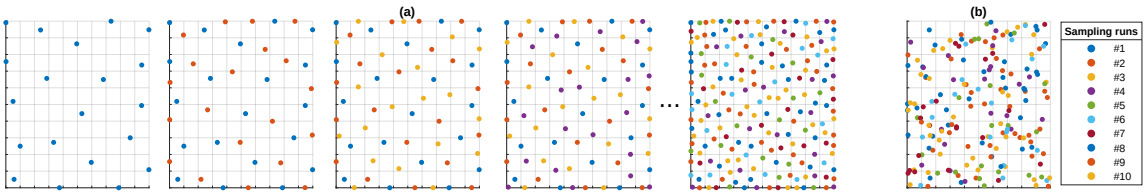


Figure 4.1: Comparison of samples generated using (a) the proposed DPS (Dynamic Propagation Sampling) and (b) MC (Monte-Carlo) approaches in a 2D design space over ten iterations.

During each iteration of Algorithm 1, the design will be generated at its step 9 using Algorithm 2. Fig. 4.1 (a) shows the results of DPS in a two-dimensional (2D) design space over ten iterations, with each iteration generating 20 designs. In the first iteration (far left image), designs are uniformly distributed and quasi-non-collapsing over the entire design space. In the second iteration (second image from the left), new samples (highlighted in orange) are not only uniformly distributed but also different from the previously sampled

Algorithm 2 The pseudo-code of sampling algorithm

function $DPS(\mathcal{X}, \hat{\mathcal{S}}, N, n_s, i_{max}, \omega)$
Input: Initialise design space ($\mathcal{X}$), samples of previous iteration ($\hat{\mathcal{S}}$), number of designs to be sampled (N), sub-population size (n_s), maximum optimisation iterations (i_{max}) and parameter ω.
Randomly create an initial population ($\mathbf{P}$) consisting of N sub-populations $\{p_1, p_2 \ldots, p_N\}$ of size n_s.
Initialise $\mathcal{S} = \{\mathbf{x}_{p_1}, \mathbf{x}_{p_2}, \ldots, \mathbf{x}_{p_N}\}$ with one design from each sub-population.
Initialise iteration count, $i \leftarrow 0$.
while $i \leq i_{max}$ **do**
  $i \leftarrow i + 1$.
  for $L = 1$ to N **do**
    for $c = 1$ to n_s **do**
      Update design $\mathbf{x}_c$ of p_L using a meta-heuristic optimiser and obtain updated design, $\mathbf{x}'_c$.
      Calculate $F_1(\mathcal{S}')$, $F_2(\mathcal{S}')$ and $F_1(\mathcal{S})$, $F_2(\mathcal{S})$, respectively, for $\mathcal{S}' = \{\mathbf{x}'_c, \mathbf{x}_{p_2}, \ldots, \mathbf{x}_{p_N}\}$ and $\mathcal{S} = \{\mathbf{x}_c, \mathbf{x}_{p_2}, \ldots, \mathbf{x}_{p_N}\}$.
      if $size(\hat{\mathcal{S}}) > 0$ **then**
        Calculate $F_3(\mathcal{S}, \hat{\mathcal{S}})$ and $F_3(\mathcal{S}', \hat{\mathcal{S}})$ with previously selected designs $\hat{\mathcal{S}} = \{\hat{x}_{p_1}, \hat{x}_{p_2}, \ldots, \hat{x}_{p_{\hat{N}}}\}$.
        $\mathbf{F}(\mathcal{S}, \hat{\mathcal{S}}) \leftarrow F_1(\mathcal{S}) + F_2(\mathcal{S}) + F_3(\mathcal{S}, \hat{\mathcal{S}})$.
        $\mathbf{F}(\mathcal{S}', \hat{\mathcal{S}}) \leftarrow F_1(\mathcal{S}') + F_2(\mathcal{S}') + F_3(\mathcal{S}', \hat{\mathcal{S}})$.
      else
        $\mathbf{F}(\mathcal{S}, \hat{\mathcal{S}}) \leftarrow F_1(\mathcal{S}) + F_2(\mathcal{S})$.
        $\mathbf{F}(\mathcal{S}', \hat{\mathcal{S}}) \leftarrow F_1(\mathcal{S}') + F_2(\mathcal{S}')$.
      end if
      if $\mathbf{F}(\mathcal{S}', \hat{\mathcal{S}}) < \mathbf{F}(\mathcal{S}, \hat{\mathcal{S}})$ **then**
        Replace the old design $\mathbf{x}_c$ with $\mathbf{x}'_c$ in p_L.
      else
        Reject the new design $\mathbf{x}'_c$ and keep $\mathbf{x}_c$ in p_L.
      end if
    end for
    Obtain the updated p_L and set as p'_L.
    Find the new best design $\mathbf{x}'_{p_L}$ from p'_L.
    Replace $\mathbf{x}_{p_L}$ with new $\mathbf{x}'_{p_L}$ in set $\mathcal{S}$ (i.e., $\mathcal{S} = \{\mathbf{x}'_{p_1}, \mathbf{x}_{p_2}, \ldots, \mathbf{x}_{p_N}\}$).
  end for
end while
return Optimal design set $\mathcal{S}$.

designs. As iterations continue, more and more regions of design space are covered, and all designs are different from each other. MC results for a similar case are shown in Fig. 4.1 (b), which clearly indicates that DPS outperforms MC in terms of sample quality.

4.2.4 Design parametrisation of free-form shapes

Contemporary CAD representations, such as B-splines and Non-Uniform Rational B-splines (NURBS), as well as their recent extensions [100–102], provide a variety of tools for constructing free-form shapes at the required level of accuracy and smoothness (fairness). For a free-from feature, depending on its complexity and the desired accuracy, single or multiple surfaces can be used, which are then knitted together to construct the entire shape. These surfaces can inherently provide local or global modification with a set of control points defining the control polygon/cage enclosing the surface. Similar to [103], one could use these control points as parameters to create design variations, but it can be challenging if the shape is composed of multiple patches; then geometric continuity between different patches must be retained; otherwise, design modification would result in invalid or irregular geometries. If geometric continuity is not the problem, then precise contraction of features requires a higher degree of surface representation, increasing the number of control points. This higher number of control points can give rise to surface irregularities during modification and can cause the curse of dimensionality during PSA and shape optimisation. Therefore, because of the aforementioned difficulties of parametrisation with control points, different specific application-dependent PD-based techniques [1,4,104] have been proposed for shape modification of free-form shapes.

In the PD context, a set of surface control points defining a particular free-form feature is coupled with a linear procedural relation creating a versatile and high-level design parameter set. For any parametric value, the procedural relations systematically modify the control net, which not only performs a plausible and fair surface modification but also ensures geometric continuity and validation of constraints. With this parametrisation, complex shapes can be parametrised with a considerably small number of parameters.

However, constructing such parametric models requires a significant understanding of the baseline design, its performance aspect and underlining geometric representation.

Unlike PD, FFD is a generic approach, originally proposed by Sederberg and Parry [61], used for parametrisation and modification of shapes and is independent of the underlying geometric representation used to create the object. This makes FFD easy to implement and efficient to create a simple transformation to achieve the desired level of design variation. FFD and its variation have been widely utilised in computer graphics [105] and in various design applications, including ship design [106].

Fig. 4.2 shows a parent hull similar to the KCS, which is extensively used in the Naval Architecture and Marine Engineering field for different studies; thus, used for experimentation in the present work. In the subsequent section, we discuss the parametrisation of this model with PD and FFD. It should be noted that both PD and FFD are implemented on the half part of the model below the waterline, mirrored along the longitudinal axis to generate the full hull form.

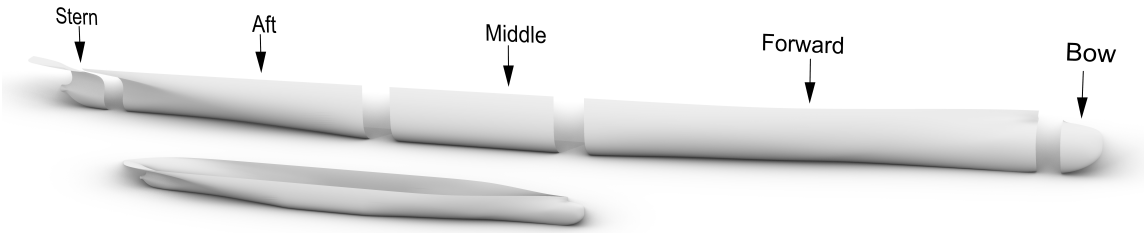


Figure 4.2: Baseline (parent) ship hull and its segmentation into different parts

Procedural deformation

The PD-based parametric modeller $\mathcal{P}(\mathbf{X})$, used in this work, is based on the technique which was introduced by Kostas et al. [1] and further developed by Katsoulis et al. [2]. In this case, $\mathcal{P}(\mathbf{X})$ is a vector function $\mathbb{R}^n \rightarrow \mathcal{A}$ that provides the geometry of the object $\mathcal{G}$, which corresponds to a user-specified parameter vector $\mathbf{X} = X_k, k = 1, 2, \ldots, 24$. If PD adopts a B-rep representation, as is the case in this work which employs T-splines for

representing surfaces, then $\mathcal{P}(\mathbf{X}) = T(CG(\mathbf{X}); \Omega)$, where $T : \mathbb{R}^2 \rightarrow \mathbb{R}^3$ is a vector-valued function, maps each point of the two-dimensional parameter domain, Ω, to a point on the surface bounding the object $\mathcal{G}$. Here, $CG(\mathbf{X})$ represents the *control cage* of the ship hull, obtained with the aid of an automatic process (ibid.), which maps $\mathbf{X}$ onto the control points of CG. The topology of CG is built in the above automatic process along with a set of internal parameters specified by the developer to accommodate the requirements implied by the chosen ship hull. $\mathbf{X}$ is a high-level parametric set containing different families of parameters of both local and non-local in nature, providing shape modification of local features (e.g. length and height of bow) and semi-global features (i.e., length and position of the parallel middle body). The graphical representation of these parameters on hull geometry is depicted in Fig. 4.3.

Apart from $\mathbf{X}$, there are also three global parameters defining the length at the water-line, beam (width) and depth of the hull. In our analysis, these three parameters are kept fixed as they are the most shape-influential parameters and are prefixed by the customers. Fixing these parameters keeps the bounding box surrounding the model fixed. $\mathbf{X}$ is defined according to the following procedural scheme:

$$X_k = \hat{X}_k \cdot f_k \left(X_1, X_2, \ldots X_{24}\right), \quad \hat{\mathbf{X}} \in [0,1],\ k = 1, 2, \ldots, 24. \tag{4.23}$$

where $\hat{X}_k$ is the k^{th} non-dimensional version of X_k and f_k is an affine function of these parameters defining the procedural relation and is specified by the developer. Like $\mathbf{X}$, $\hat{\mathbf{X}}$ is also an external parametric set and is bounded by $[0, 1]$. During design modification, [1, 2] recommend the usage of $\hat{\mathbf{X}}$ by the user as these parameters support the robustness by avoiding setting the parametric values that would result in creating invalid and implausible geometries. Once the values of global parameters and $\hat{\mathbf{X}}$ are provided, the control cage for each of the five hull segments shown in Fig. 4.2 can be constructed. For further details on the formulation of this approach, interested readers should refer to [1, 2].

Note that in Fig. 4.3, along with $X_k, k = 13, 14$, there are two more parameters, X'_k,

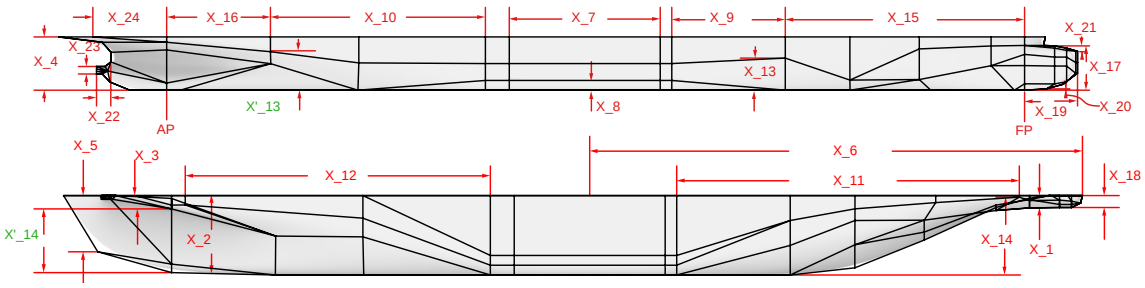


Figure 4.3: Ship-hull parametrisation adopted by the PD-based parametric modeller. Sensitivity analysis is performed only for the parameters depicted in red and parameters in green, X'_{13} and X'_{14}, linearly depend on X_{13} and X_{14}.

$k = 13, 14$ (highlighted in green), for the forward and the aft segment of the hull. However, each pair $\{X_k, X'_k\}$ is linearly dependent, i.e., $X'_k = p_{int,k} X_k$, where $p_{int,k}, k = 13, 14$, is an internal parameter, decided by the developer and kept fixed for a given materialisation of PD and thus X'_k, $k = 13, 14$ are not included in the final parametric set for which sensitivity and intra-sensitivity are evaluated.

Free-form deformation

FFD is composed of a set of control points that belong to the trivariate parametric hyper-patch. It encloses the free-form shape to be deformed and is also referred to as the control volume of FFD. The variation of lattice control points induces modification in the embedded shape. In analogy to the control net of splines, the deformation is governed by the control volume, and depending on the underlying basis, both local and global deformation can be achieved. For detailed formulation on FFD, interested readers should refer to [61].

The geometric representation of design parameters of the parent hull created with FFD is shown in Fig. 4.4. In this case, the hull is divided into four segments, namely bow, forward, middle and aft, instead of segmentation into five parts employed in the PD case. More specifically, the PD stern and aft segments are merged into a single one in the FFD case. Otherwise, a separate stern segment would require a separate control cage, substantially increasing the number of control points needed to define the shape.

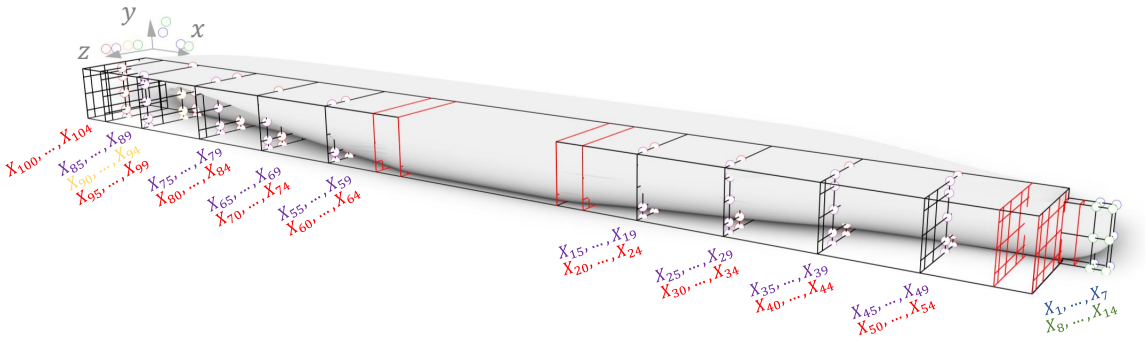


Figure 4.4: Ship-hull parametrisation based on FFD (Free-Form Deformation). Layers of the lattice cage depicted in red are fixed to maintain continuity between the connecting segments. Moreover, the control points marked in red, purple and yellow only move in the z-direction, whereas control points in green can move in all three directions, and blue ones move in x- and y-directions.

The lattice cage is constructed separately for each segment of the hull in Fig. 4.2. The forward and aft segments consist of eight planar layers, each consisting of $[5 \times 6]$ control points, whereas the bow consists of four layers, each containing $[3 \times 2]$ control points. To maintain the G^0 and G^1 geometric continuity between segments during deformation, the two layers of each segment neighbouring to the connecting point of the other segment are kept fixed, which are depicted in red in Fig. 4.4. The control points close to the hull surface in the forward and aft segment are moved in the z-direction as they significantly affect the deformation compared to the other control points. The movable control points in each segment of the hull are indicated in Fig. 4.4. The selection of movable control points is made to ensure that they modify the similar feature as of PD, creating the same ground for comparison of both techniques. Furthermore, the modification at the middle body segment is not performed to satisfy the bounding box constraint. However, changing the parameters, X_{15} to X_{79}, enables this parameterisation to render the hull shape more cylindrical and thus change the length and position of the midship part.

4.2.5 Assessing the impact of sensitive parameters on geometry

As mentioned earlier, unlike solid models resulting from concurrent modelling, the parametrisation of free-form shapes is not fully feature-driven. The impact of the change of parameters on the shape can be challenging to quantify precisely how much a particular region of the free-form shape will deviate when any design parameter is altered. Therefore, once a set of sensitive parameters is obtained, it is essential to estimate the impact of these parameters on the free-form shape to identify the geometrical features sensitive to the QoI. Moreover, identifying these features can also facilitate the comparison of different parametric modellers, which may be different with respect to the adopted design space and its dimensionality, as in the case of PD and FFD.

To identify these features, we propose an approach similar to [3,107], which uses Hausdorff distance to compute the geometric difference between the two free-form shapes. In the present case, hull form parametrised with PD and FFD is composed of T-spline and NURBS surfaces, respectively; therefore, before evaluating the Hausdorff distance, it should be discretised either by sampling points on it or by creating a mesh, in which vertices are used as points to measure Hausdorff distance.

Consider the value of a k^{th} design parameter, X_k, of baseline design, $\mathcal{G}$, is changed from x_k to x'_k and is inputted to the parametric modeller, which creates a new design $\mathcal{G}'$. Afterwards, both $\mathcal{G}$ and $\mathcal{G}'$ are discretised to create point sets, $\mathbf{O} = \{\mathbf{o}_i, \forall i \in \{1, 2, \ldots, n_o\}\} \in \mathcal{G} \subseteq \mathbb{R}^3$ and $\mathbf{O'} = \{\mathbf{o}'_j, \forall j \in \{1, 2, \ldots, n'_o\}\} \in \mathcal{G}' \subseteq \mathbb{R}^3$, respectively, containing total n_o and n'_o points. Now, to identify the regions of the design which are modified by changing x_k to x'_k and to measure the extent of this modification, we evaluate Hausdorff distance, E, between $\mathbf{O}$ and $\mathbf{O'}$ as

$$E(\mathcal{G}, \mathcal{G}') = max(e(\mathbf{o}_i, \mathcal{G}'),\ \forall i \in \{1, 2, \ldots, n_o\}), \tag{4.24a}$$

$$e(\mathbf{o}_i, \mathcal{G}') = min(\mathcal{M}(\mathbf{o}_i, \mathbf{o}'_j),\ \forall j \in \{1, 2, \ldots, n'_o\}), \tag{4.24b}$$

$$\mathcal{M}(\mathbf{o}_i, \mathbf{o}'_j) = \sqrt{\sum_{t=1}^{3}(o_{t,i} - o'_{t,j})^2}, \tag{4.24c}$$

where, $E(\mathcal{G}, \mathcal{G}')$ is the one-sided Hausdorff distance between $\mathcal{G}$ and $\mathcal{G}'$, $\mathcal{M}(\mathbf{o}_i, \mathbf{o}'_j)$ is the Euclidean norm between two points and $e(\mathbf{o}_i, \mathcal{G}')$ is minimum distance between the i^{th} point $\mathbf{o}_i \in \mathbb{R}^3$ on $\mathcal{G}$ and all the n'_o points $(\mathbf{o}_j, \forall j \in \{1, 2, \ldots, n'_o\} \in \mathbb{R}^3)$ on $\mathcal{G}'$. It should be noted that Hausdorff distance is not symmetrical (i.e., $E(\mathcal{G}, \mathcal{G}') \neq E(\mathcal{G}', \mathcal{G})$). Therefore, a two-sided Hausdorff distance, $\mathcal{E}(X_k)$, is evaluated as

$$\mathcal{E}(X_k) = max(E(\mathcal{G}, \mathcal{G}'), E(\mathcal{G}', \mathcal{G})). \tag{4.25}$$

As we know, each design parameter is bounded by the upper and lower limit. Therefore, x_k can be varied between its parametric range (i.e., $x'_k \in [x^l_k, x^u_k]$), and we are interested in identifying x'_k that results in maximum deviation of $\mathcal{G}'$ from $\mathcal{G}$. Hence, $\mathcal{E}(X_k)$ is evaluated over the entire range of X_k as

$$\mathcal{E}_{max}(X_k) = max(E_{max}(\mathcal{G}, \mathcal{G}'), E_{max}(\mathcal{G}', \mathcal{G})), \tag{4.26a}$$

$$E_{max}(\mathcal{G}, \mathcal{G}') = max(E(\mathcal{G}, \mathcal{P}(x'_k)),\ \forall x'_k \in [x^l_k, x^u_k]), \tag{4.26b}$$

$$E_{max}(\mathcal{G}', \mathcal{G}) = max(E(\mathcal{P}(x'_k), \mathcal{G}),\ \forall x'_k \in [x^l_k, x^u_k]). \tag{4.26c}$$

Once the realisation x'_k giving $\mathcal{E}_{max}(X_k)$ is obtained then at this value Eq. (4.24b) can be evaluated for all n_o points of $\mathcal{G}$ (i.e., $\mathbf{e} = e(\mathbf{o}_i, \mathcal{G}'),\ \forall i \in \{1, 2, \ldots, n_o\}$). Afterwards, these values are projected on the points of $\mathcal{G}$ and the feature saliency map is created, which helps to visually identify the region of design that is modified by changing X_k. An example of this map is shown in Fig. 4.5. Once $\mathcal{E}_{max}(X_k)$ is obtained for all the parameters in $\mathbf{X}$, then a parameter with the highest value of $\mathcal{E}_{max}(X_k)$ can be regarded as most shape influential as it has the highest impact on the modification of the baseline design.

In this work, $\mathbf{O}$ consists of the vertices of a triangular mesh over the bounding surface

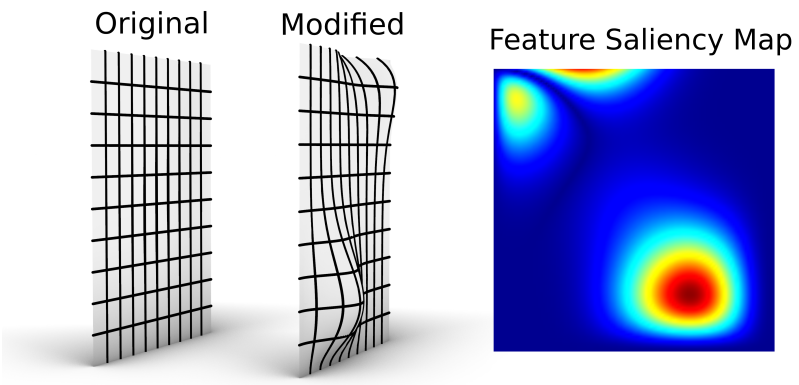


Figure 4.5: Illustration of the *features saliency map* between two free-form surfaces: original surface (left image), modified surface (middle image), the feature saliency map (right image) with red regions indicating the maximum deviation of the modified from the original surface.

of $\mathcal{G}$. This mesh is created using Rhinoscript's MESH function, which, on a PC with a Xeon(R) Gold 6226, 2.7-GHz processor, and 128-GB memory, takes 0.84, 1.83 and 8.47 seconds for Rhinoceros®[3] 3D to mesh the parent hull surface with $n_o = 39,823$, $246,023$ and $2,228,224$ vertex points, respectively. This indicates that the computational cost of mesh generation is very close to a linear function of n_o.

4.3 Results and discussion

In this section, we first discuss the experimental settings used for the verification of the proposed approach. Afterwards, under these settings, the results of sensitivity analysis performed with ASM are presented along with its proposed regional version to determine the intra-sensitivity of parent hull parametrised with PD and FFD. The features of the parent hull corresponding to the sensitive and intra-sensitive parameters are evaluated, and a correlation between the results of PD and FFD is studied. Finally, the performance of the Algorithm 1 and 2 is compared with Sobol's sensitivity, CSV and MC sampling

[3]https://www.rhino3d.com/

4.3.1 Experimental configuration

The proposed method is tested for a parent hull parametrised with PD and FFD, respectively, creating a design space with $n = 24$ and $n = 104$ dimensions. The volume of displacement (∇) and total resistance (R_T) of the model are taken as QoIs. Here, ∇ is the volume of the water displaced by the ship, which is purely an associated geometrical quantity and R_T is the total resistance of the ship travelling with constant velocity on the otherwise clam free surface of the ocean. Both quantities are of critical interest during all phases of ship design. Thus, it is typical that the designer aims to reduce R_T while keeping ∇, which is essentially the displacement, close to the baseline design. The three basic components of the R_T are frictional resistance (R_F), wave-making resistance (R_W) and viscous pressure resistance (R_V). In this study, R_T is evaluated using the well-known regression method of Holtrop and Mennen [108], which is widely used at the preliminary design stage. The main particulars and R_T value for the parent hull are given in Table 4.1.

Table 4.1: Parent container-ship hull: main particulars and total resistance R_T

Quantity	Symbol	Unit	Value
Volume of displacement	∇	m^3	44868.95
Wetted surface area	S	m^2	8293.65
Length at waterline.	Lwl	m	200
Beam at waterline	Bwl	m	32.20
Draft	T	m	10.80
Water density	ρ	kg/m^3	1025
Kinematic viscosity	ν	m^2/s	1.09E-6
Gravity acceleration	g	m/s^2	9.803
Speed	V	knots	25
Total resistance	R_T	N	2.6374e+06
Total resistance coefficient	C_T	-	0.0038

The design spaces for both parametrisation methods are created using the strategy mentioned in §3.1.1. In order to create the same ground for comparing PD and FFD, the

parametric ranges are set in a way that both parameterisations provide similar variability for ∇ and R_T. The PDF of ∇ of the designs sampled from both design spaces are shown in Fig. 4.6. From these results, it can be seen that for both design spaces, the distribution of ∇ is similar. The statistical results, given in Table 4.2, also show that the ∇ of designs sampled from PD and FFD-based design spaces have a similar standard deviation, mean, maximum and minimum values.

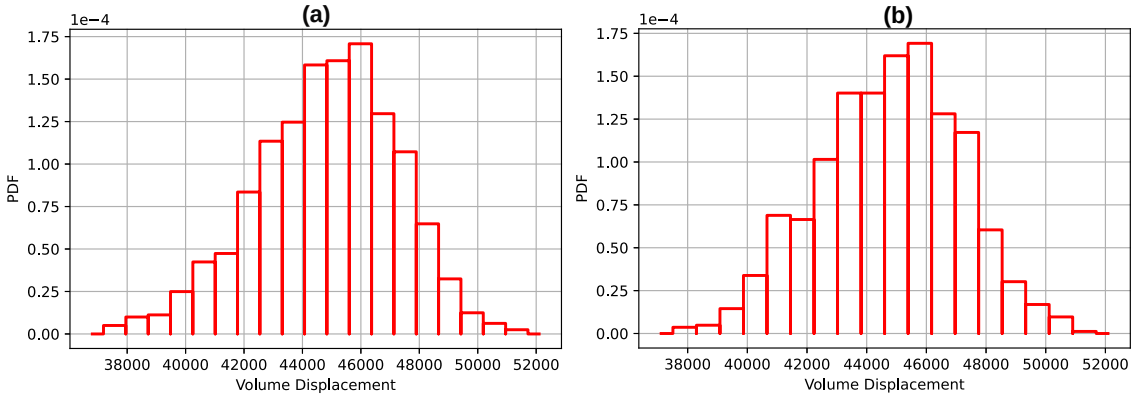


Figure 4.6: PDF (Probability Density Distribution) of ∇ (Volume of Displacement) of designs sampled from a design space created with (a) PD (Procedural Deformation) and (b) FFD (Free-Form Deformation).

Table 4.2: Statistics of ∇ from samples of the design space created with PD and FFD.

	PD	FFD
Standard deviation	2398.42	2351.92
Mean	44827.21	44767.22
Maximum	51718.96	51691.54
Minimum	37189.46	37506.49

4.3.2 Sensitivity analysis of hull parametrised with PD

The evaluation of parameters' sensitivity indices for ∇ and R_T commences with Algorithm 1 which takes $\mathcal{X}$, $\mathcal{G}$, $\mathcal{P}$, g and ℓ as inputs. Then iterations are performed until the total number of samples, N, reaches the allowable number of samples, $N_{allow} = 2000$ or sensi-

tivity ranking of parameters gets stable for 5 consecutive iterations. During each iteration of Algorithm 1, DPS is performed with 10 optimisation iterations to create a sample set, $\mathcal{S}$, containing N designs sampled from $\mathcal{X}$. At the first iteration γ in $N = \gamma n \log(n)$ is set $\gamma = 1.31$, which resulted in $N = 100$ samples for first iteration. In the subsequent iterations N is half of its initial value. Furthermore, following [38], DPS optimisation parameters, n_s and ω, are set to n and $n/2$, respectively.

Once sampling is completed, the samples are fed to the PD-based parametric modeller to create modified instances. Afterwards, ∇ and R_T of these instances are obtained along with their gradients, estimated with the local-surrogate model constructed with a subset containing $\ell = 60$ samples, to construct the covariance matrices. The eigendecomposition of these matrices provides the corresponding eigenvalues and eigenvectors. The former defines a separation measure for active and inactive subspace while the latter span their bases. Then sensitivity analysis is performed in the active subspace to assess the global sensitivity indices of parameters in $\mathbf{X}$. The eigenvalue plots for ∇ and R_T are shown in Fig. 4.7 (a).

For both ∇ and R_T, these plots reveal a large gap between the first and the second eigenvalues, which shows strong potential for one-dimensional ($m = 1$) active subspaces. The sensitivity indices, evaluated via Eq. (4.7), for ∇ and R_T, are shown in Fig. 4.7 (b). From these results it can be observed that in case of ∇ the parameter, X_9, is the most sensitive one followed by X_{15}, X_{10}, X_7 and X_{14}, whereas X_1, X_3 and X_{17} to X_{23}, show negligible effect. In case of R_T, parameters, X_{10}, X_9 and X_7, are the three most significant followed by, X_{15}, X_{12}, X_{16} and x_{17}, with relatively lower impact. It is interesting to note that parameter, X_{17}, which affects the bulbous bow height, had the lowest impact on ∇ but have a relatively high influence on R_T, which can be attributed due to the fact that the bulbous bow plays a significant role in reducing R_T.

Fig. 4.8 shows the convergence plot for the sensitivity index, $\boldsymbol{\alpha}$, and ranking, $\mathbf{R}$, versus the number of designs over the first 20 iterations of Algorithm 1. It can be observed that the sensitivity indices and the ranking of the sensitive parameters tend to be stable

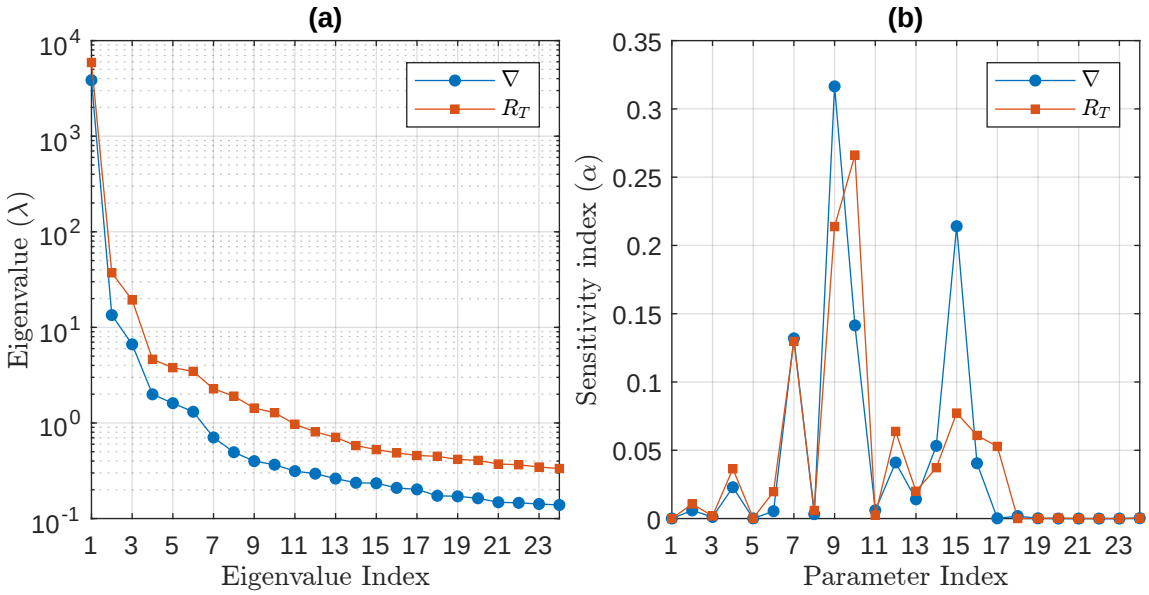


Figure 4.7: (a) Eigenvalues and (b) Sensitivity indices of all 24 PD (Procedural Deformation) parameters obtained from Eq. (4.3) and (4.7), respectively, when ∇ (Volume of Displacement) and R_T (Total Resistance) are used as QoI (Quantities of Interest).

even after the first few iterations. For instance, in case of ∇, parameters, X_9, X_{15}, X_{10} X_7 and X_{14}, become stable after 7^{th} iteration at $N = 400$. Similarly, highly significant parameters for R_T, namely X_{10}, X_9, X_7, X_{15} and X_{12}, achieve stability after the 3^{rd} iteration. This reveals that the stability of the parametric ranking with respect to the sample size is directly proportional to their sensitivity indices. Meaning, a parameter with a high sensitivity index is less affected by the sample size than an insensitive one. The main reason for such a behaviour is that sensitivity indices of the least sensitive parameters, $\{X_1, X_2, X_3, X_{18}, \ldots, X_{24}\}$, are very close to each other so, as explained earlier, a slight change in their indices changes their ranking. Therefore, it requires a large number of samples to stabilise the ranking of these parameters. It is worthy to point out that a designer is not always interested in the right ranking of parameters as the decision on characterising parameters as significant or insignificant is made based on sensitivity indices not on the ranking [31]. Therefore, to save the computational budget, Algorithm 1 can be

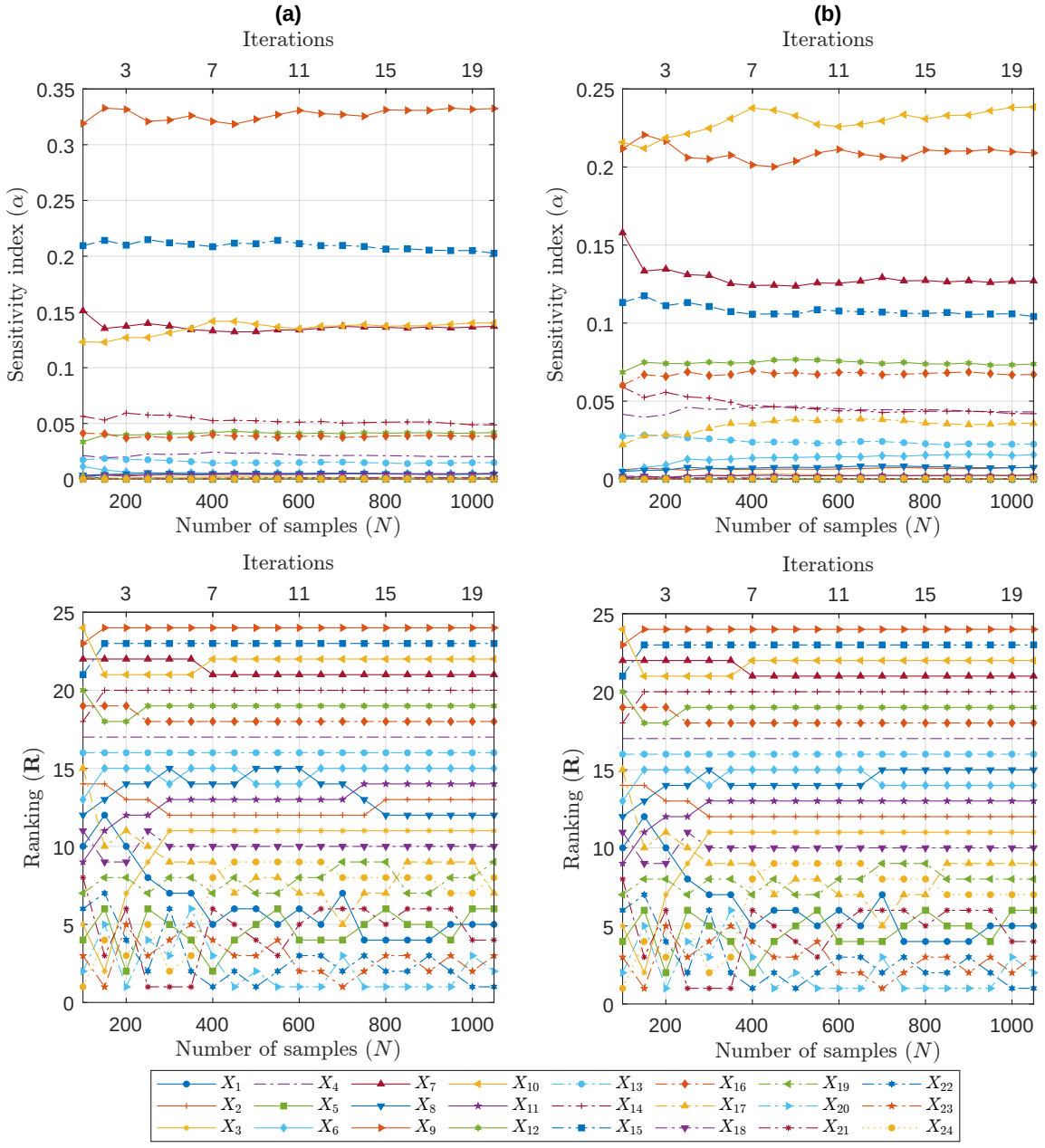


Figure 4.8: Convergence plots of sensitive indices and parametric ranking of all 24 PD parameters for (a) ∇ (Volume of Displacement) and (b) R_T versus the number of samples over the first 20 iterations of Algorithm 1 when it is used in conjunction with DPS (Dynamic Propagation Sampling).

terminated as soon as the ranking of the significant parameters stabilises.

Regional analysis with PD

Using the setting already set in this section we proceed with the regional analysis for ∇ and R_T. Here, the formulation and the detailed results for ∇ are discussed first and then a condensed summary of the results of R_T are provided. To start with, Fig. 4.9 depicts 2D regional sensitivity indices of ∇ with fixed upper limit and varied lower limit, $\alpha_{k'}(q_k^l)$ (highlighted in blue), and vice versa $\alpha_{k'}(q_k^u)$ (highlighted in orange), for all 24 parameters obtained using Eq. (4.13). As it can be easily seen from this Figure that for some parameters these indices have a monotonic trend while for other parameters they exhibit a flat behaviour over their parametric range. It is noteworthy that the sensitive parameters, X_7, X_9, X_{10}, X_{14} and X_{15}, follow a monotonic behaviour as their range shrinks. A rather strong decline in $\alpha_{k'}(q_k^l)$ and $\alpha_{k'}(q_k^u)$ is observed within the sub-range $[q_k^l = 0, q_k^u = 0.5]$ and $[q_k^l = 0.5, q_k^u = 1.0]$, respectively, which gradually slows down after $q_k^l, q_k^u = 0.5$.

Furthermore, parameters with a sensitivity index close to 0.05, such as X_4, X_{12} and X_{16}, also show averagely a monotonic but non-smooth behaviour over their range. The parameters with negligible ($\ll 0.05$) value of sensitivity index are insensitive with respect to the reduction to their range. This shows that the higher the sensitivity index is, the probability of having a monotonic and smooth behaviour of $\alpha_{k'}(q_k^l)$ and $\alpha_{k'}(q_k^u)$ over their range is also high.

Fig. 4.10 shows the 3D regional sensitivity indices, $\alpha_{k'}(q_k^{l,u})$, obtained over the entire parametric range for ∇ using Eq. (4.16). To analyse these plots we provide some auxiliary remarks. In the lower triangle of the unit square we have $\alpha_{k'}(q_k^{l,u}) = 0$ as the result of the fact that q_k^u and q_k^l do not obey the constraint, $q_k^u > q_k^l$. Moreover, along the counter diagonal of the unit square (i.e., the line connecting $[q_k^l = 0, q_k^u = 1]$ and $[q_k^l = 1, q_k^u = 0]$) the range of X_k reduces symmetrically, e.g., from $[q_k^l = 0, q_k^u = 1]$ to $[q_k^l = 0.1, q_k^u = 0.9]$. In the lower triangle of the unit square, any point on a line parallel to the diagonal has the

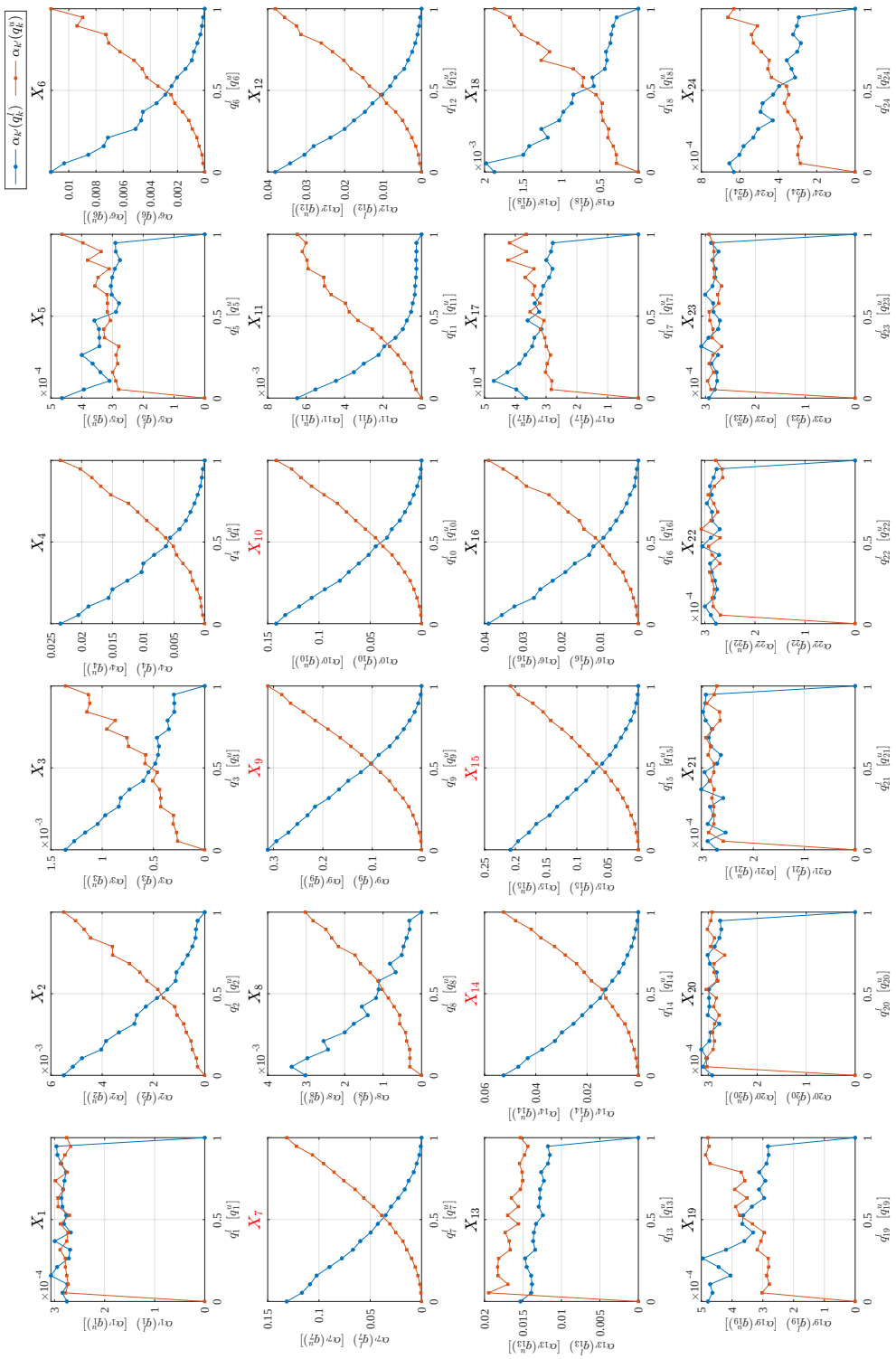


Figure 4.9: 2D regional-sensitivity plots of all 24 PD (Procedural Deformation) parameters when ∇ (Volume of Displacement) is used as QoI (Quantity of Interest) and the range of each parameter is varied with fixed upper and varied lower limit and vice versa. The parameters highlighted in red are the top most sensitive parameters.

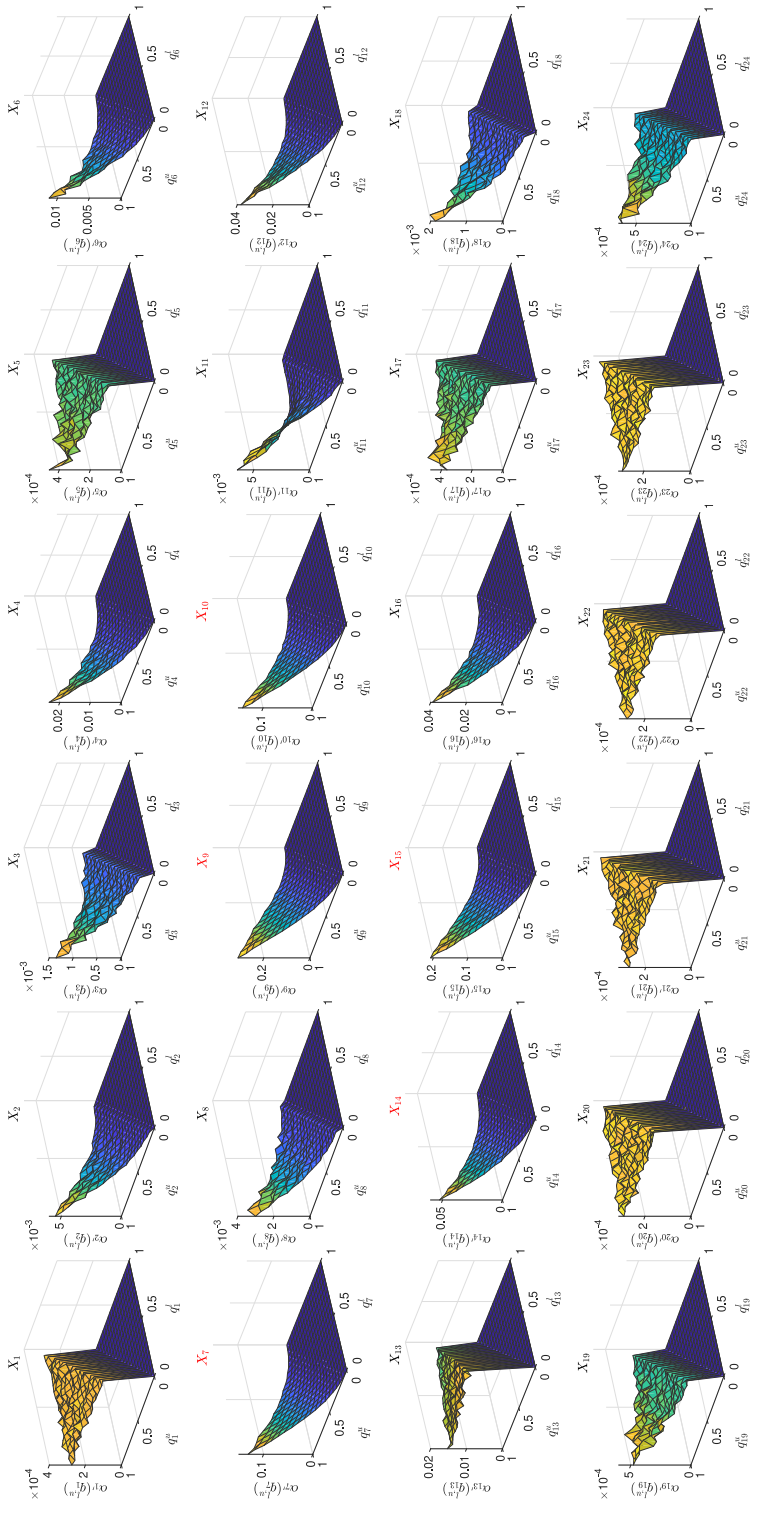


Figure 4.10: 3D regional-sensitivity plots of all 24 PD parameters when ∇ is used as QoI and the range of each parameter is varied. The parameters highlighted in red shows the plots of the top most sensitive parameters.

same range length, i.e., $|q_k^u - q_k^l|$ is the same, nevertheless, $\alpha_{k'}(q_k^{l,u})$ is not constant along this line. For instance, in case of X_{10}, $\alpha_{9'}(q_{10}^{l,u})$ is equal to 0.01175 for $[q_k^l = 0.94736, q_k^u = 1]$, whereas it is equal to 0.01946 for $[q_k^l = 0, q_k^u = 0.05264]$. Even though both intervals $[q_k^l = 0.94736, q_k^u = 1]$ and $[q_k^l = 0, q_k^u = 0.05264]$ share a common length, but they have different location over $[0, 1]$ and, as implied by Figure 10, their sensitivity indices are different. In addition to that these plots also accumulate the information of 2D plots in Fig. 4.9. The boundary of the lower triangle parallel to the q_k^l axis shows the values of $\alpha_{k'}(q_k^l)$, likewise the boundary parallel to q_k^u axis show $\alpha_{k'}(q_k^u)$.

Fig. 4.11 shows the 3D regional sensitivity plots for $\alpha_{k'}(q_9^{l,u})$, with $k' = 1, 2, \ldots, 24$, when ∇ is used as QoI. These plots show the effect of shrinkage in the range of X_9 on the sensitivity of the remaining parameters. It can be seen that the sensitivity of all parameters changes as the range of X_9 varies. This variational effect is small on some parameters, such as X_1, X_5, X_{17} to X_{23}, while substantial on others. Except from $\alpha_{6'}(q_9^{l,u})$ for X_6, all the remaining parameters show a similar but an interesting trend. The value of all $\alpha_{k'}(q_9^{l,u})$ is relatively constant along q_9^u specially at the boundary and it tends to vary along q_9^l, however, for X_6 this increment is noticeable along both q_9^u and q_9^l.

When the range of X_9 changes from $[q_k^l = 0, q_k^u = 1]$ to $[q_k^l = 0, q_k^u = 0.6842]$, the sensitivity ranking for ∇ also changes from $X_9 > X_{15} > X_{10} > X_7$ to $X_{15} > X_7 > X_9 > X_{10}$. This shows that, within the local region defined by the sub-range, $[q_k^l = 0, q_k^u = 0.6842]$, the parameter X_9 has low sensitivity impact on ∇ compare to X_{15} and X_7. This shows that, the sensitivity of a parameter depends on how a design space is set. For any parameter, its sensitivity does not only depend on its own parametric range but also on the range of the remaining parameters. Here, we are mainly interested in this local variational behaviour of parameters' sensitivity. To visualise this we proposed to use *variational sensitivity plots*, as shown in Fig. 4.12 for all 24 design parameters. These plots accumulate the key information related to all 24 3D plots, used to study the regional sensitivity of a certain parameter, into a single 2D plot. For a k^{th} parameter, X_k, the variational sensitivity plot shows maximum, $\alpha_{k'}^{max}(q_k^{l,u})$ (indicated in black), and

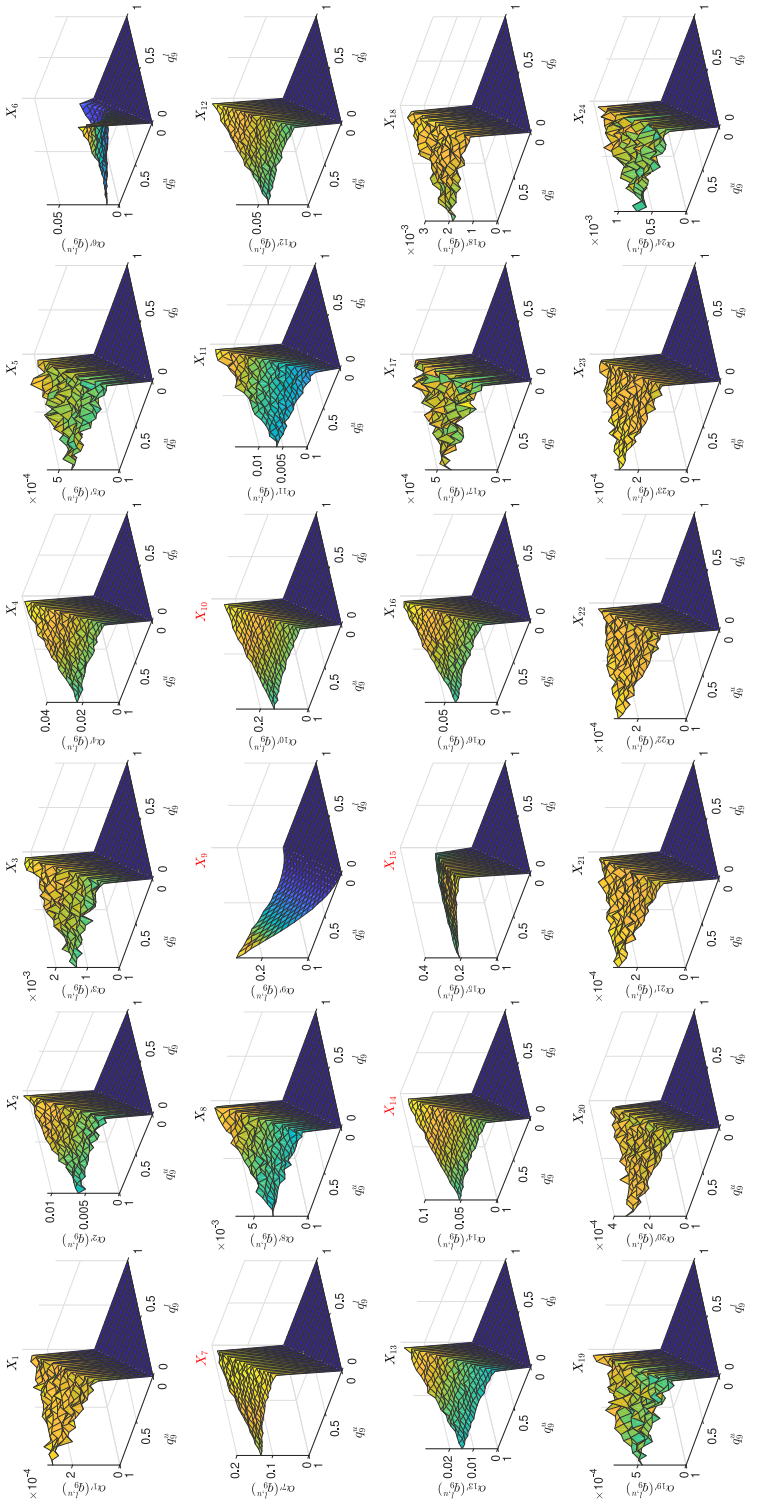


Figure 4.11: 3D regional-sensitivity plots of all 24 PD parameters in case of ∇ when the range of parameter, X_9, is varied. The red highlighted parameters are the top most sensitive ones.

minimum, $\alpha_{k'}^{min}(q_k^{l,u})$ (indicated in red), index values, which are evaluated using Eq. (4.17b) and (4.17c), respectively. More importantly these plots also includes the variability between maximum and minimum values of $\alpha_{k'}(q_k^{l,u})$ (highlighted in cyan) for all parameters when range of X_k varies between $q_k^l, q_k^u \in [0, 1]$.

A preliminary insight on the results of Fig. 4.12 shows that any perturbation in the range of X_6, X_9, X_{10} and X_{15} induces high variability in the sensitivity of all parameters (see cyan bars in the plots of Fig. 4.12). Interestingly, among these parameters, X_6 is the least sensitive one but still shows a profound impact on other parameters' sensitivity, which is prominent on X_9, X_{10} and X_{15}. Furthermore, between X_7 and X_{16}, the parameter X_7, despite having higher sensitivity index, have low effect on the variation of $\alpha_{k'}(q_7^{l,u})$. Similar, behaviour can be observed for the results of R_T given in Fig. 4.13. Along with the most sensitive parameters, X_{10}, X_9, X_{15}, the change in range of one of the least sensitive parameters, X_6, greatly influences the sensitivity of the remaining parameters.

To conclude, the results of regional sensitivity obtained in this subsection can not only facilitate designers in analysing the behaviour of the sensitivity within a certain region of the design space but, along with intra-sensitivity, also help to make informed decisions in setting the parametric ranges to create a viable design space.

Intra-sensitivity of PD parameters

Fig. 4.14 shows the intra-sensitivity index, I_k, defined in Eq. (4.12), of the PD parameters for both ∇ and R_T. It can be seen that despite being the least sensitive parameter, X_6 has the highest intra-sensitivity index for both ∇ and R_T that is the sensitivity of other parameters is highly affected by its parametric range. The next three most intra-sensitive parameters are X_9, X_{10} and X_{15} for both ∇ and R_T. Furthermore, X_7, which is the fourth/third sensitive parameter for ∇/R_T, respectively, is sixth in the intra-sensitive ranking. As for the remaining sensitive parameters, such as X_{12}, X_{14}, X_{17}, they do not show any significant intra-sensitivity. This shows that a sensitive parameter is not neces-

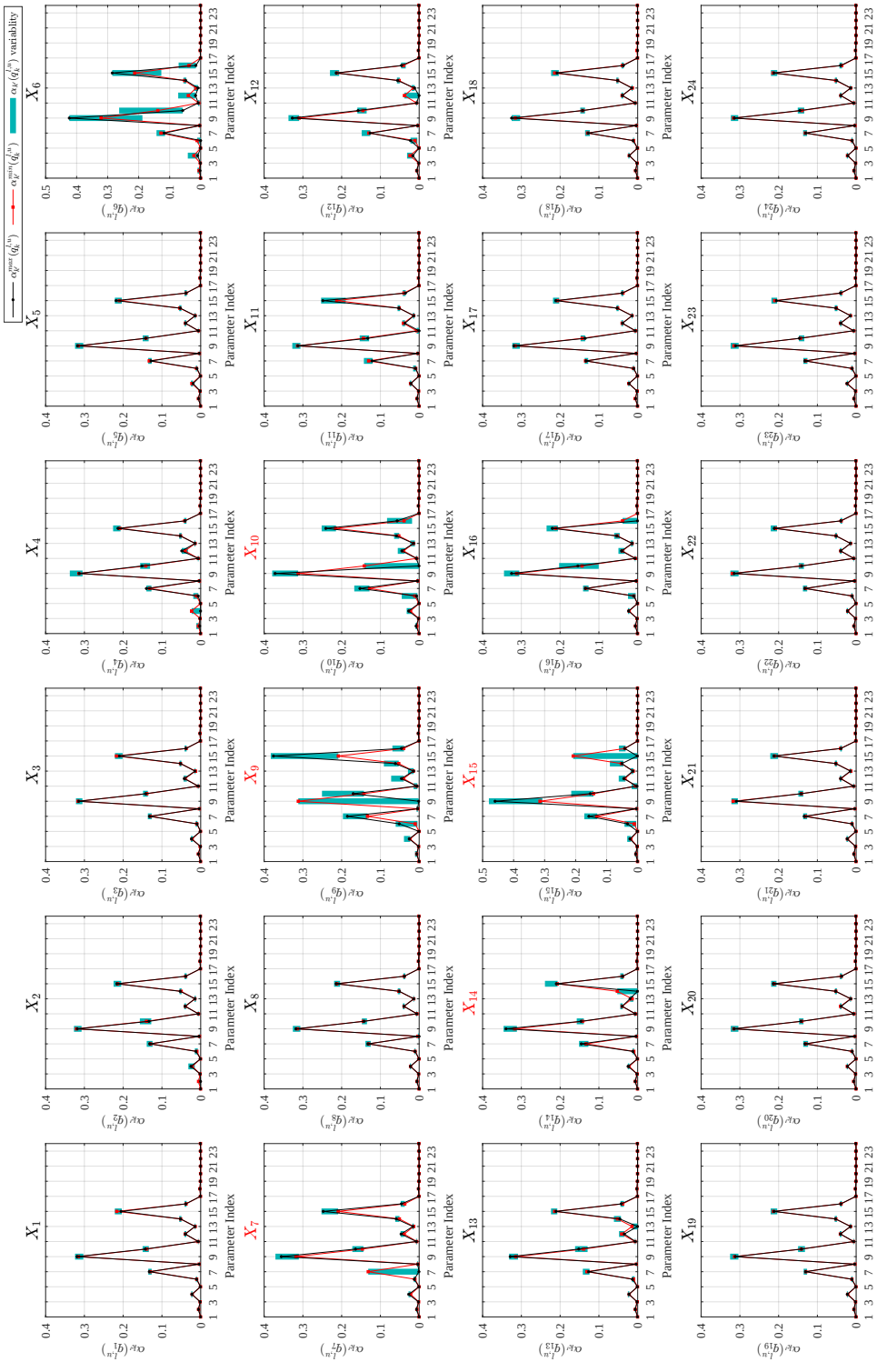


Figure 4.12: Variational sensitivity plots of all 24 PD (Procedural Deformation) parameters when ∇ (Volume of Displacement) is used as QoI (Quantity of Interest). The parameters highlighted in red are the top most sensitive parameters.

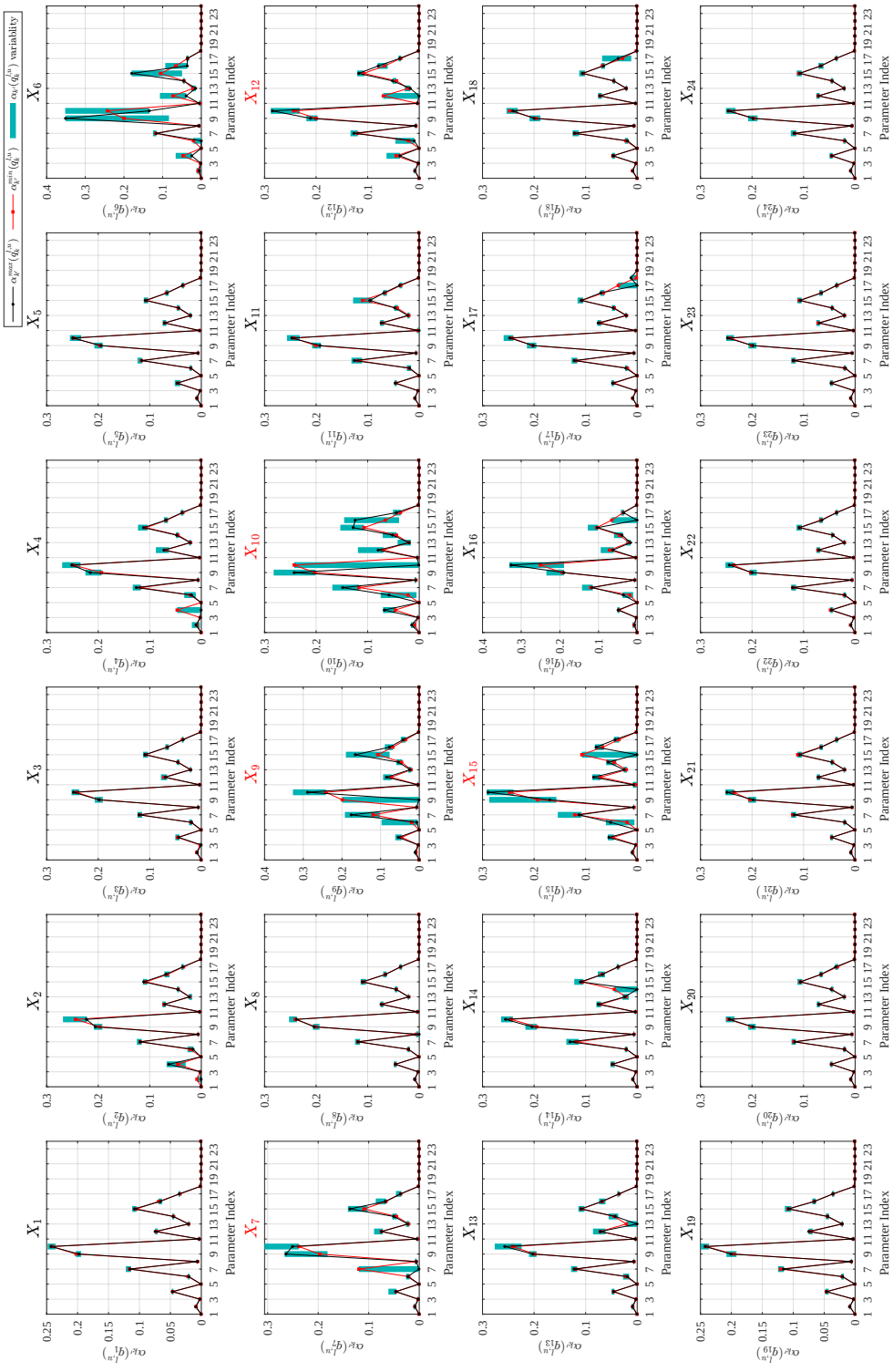


Figure 4.13: Variational sensitivity plots of all 24 PD parameters when R_T is used as QoI. The red highlighted parameters show the plots of the top most sensitive ones.

sarily an intra-sensitive one or vice versa.

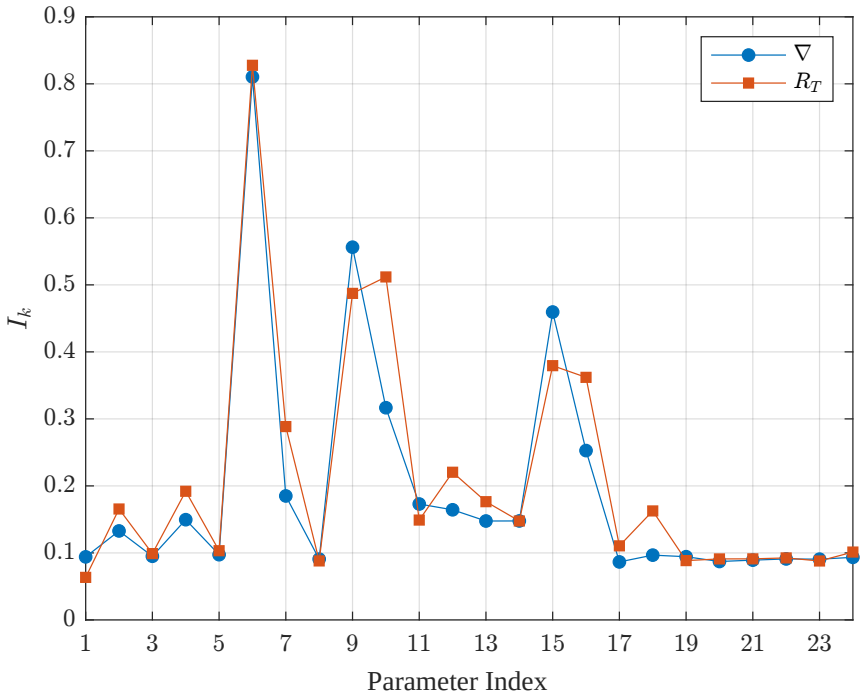


Figure 4.14: Intra-sensitivity of all 24 PD (Procedural Deformation) parameters obtained from Eq. (4.17) when ∇ (Volume of Displacement) and R_T (Total Resistance) are used as QoI (Quantity of Interest).

Any change in the range of intra-sensitive parameter significantly affects the sensitivity indices of the other parameters, thereby, altering their ranking. For instance, in case of ∇ at $[q_6^l = 0, q_6^u = 1]$ the ranking of parameters is $X_9 > X_{15} > X_{10} > X_7 > X_{14} > X_{16}$. When the range of X_6 is perturbed to $[q_6^l = 0.2105, q_6^u = 0.8947]$ or $[q_6^l = 0.4211, q_6^u = 0.6316]$ the ranking changes to $X_{10} > X_{15} > X_9 > X_7 > X_{14} > X_{16}$ or $X_{15} > X_9 > X_7 > X_{12} > X_{14} > X_{16}$, respectively. It is interesting to note that at $[q_6^l = 0.4211, q_6^u = 0.6316]$ the parameter X_{10}, which is the fourth most sensitive parameter over the original range, is not in the group of top-six sensitive parameters. Therefore, studying only parametric sensitivity is not enough especially at the preliminary design stage. This is due to the fact that at this stage designer is not knowledgeable of the appropriate parametric ranges for performing a reliable sensitivity study; thus, the sensitivity results may not align with the

designer's intuition regarding the sensitivity of parameters. Therefore, measuring the intra-sensitivity of parameters is essential for viable design space. As explained earlier, designers commonly use sensitivity analysis either for design space dimensionality reduction via fixing low sensitive parameters or for uncertainty reduction to reduce QoI's variability. Therefore, sensitivity analysis within a non-viable design space can be dismaying; either resulting in the elimination of an important parameter from the design study or in wastage of computational resources if uncertainty reduction is performed with inaccurately estimated sensitive parameters.

The results of the intra-sensitivity analysis can help not only to evaluate the specific parametric interval, within which a parameter can have a high or low sensitivity but also identifies a parameter whose variation in range notably affect the sensitivity ranking. Consequently, during the design analysis, the results of intra-sensitivity can facilitate designers in tuning an existing design space in such a way that a specific parameter becomes more significant compare to other parameters. Furthermore, care can be taken while setting the parametric limit of the most intra-sensitive parameter so that variation in range does not highly affect the parameters' ranking. For instance, in a certain design study, a parameter is significant for a certain criterion, which, let us say, is associated with the manufacturing process. Now the designer intends to include this parameter in the list of final significant parameters when sensitivity analysis is performed with respect to the design's performance. However, for any specific setting of the design space, this parameter may be insignificant with respect to performance. Therefore, to make it significant from both manufacturing and performance point of view, the designer must also perform intra-sensitivity analysis. Then, based on the obtained results from this analysis they should refine the design space so that this parameter becomes significant for both criteria.

4.3.3 Sensitivity analyses of hull parameterised with FFD

Similar to PD, the sensitivity index of FFD parameters for ∇ and R_T are evaluated with the Algorithm 1. For the economy of the presentation, we shall discuss the results of ∇

only. Algorithm 1 initiates by setting $\gamma = 1.242$, which results in $N = 600$ samples for the first iteration and N is equal to 300 for the subsequent. In this case, the number of parameters is $n = 104$; thus, in comparison to PD, FFD requires a larger number of samples resulting in the high computational cost of Algorithm 1 and 2. To evaluate the model-based gradients, the size of the subset is set to $\ell = 200$. The results of eigendecomposition for ∇ are shown in Fig. 4.15 (a), which reveals a potential for a one-dimensional active subspace. The global sensitivity indices for ∇ obtained from this active subspace using Eq. (4.7) are shown in Fig. 4.15 (b).

It can be seen that parameters, X_{31}, X_{28} and X_{41}, are the three most sensitive parameters with similar indices followed by X_{38}, X_{42} and X_{76}. Among these six parameters, X_{76} affects the aft segment of the hull while the reaming ones influence the forward segment. This means that, in term of ∇, the forward segment is more significant. These results also align with the results of PD in which among the top three significant parameters, X_9, X_{15} and X_{10}, the first two, X_9 and X_{15}, are related to the forward segment of the hull.

As the design space in FFD has a significantly higher dimension in comparison to PD ($n = 24$ for PD and 104 for FFD), therefore, the visual interpretation of the sensitivity results is more challenging. To ease analyses of such high dimensional problem, Sheikholeslami et al. [37] recommend clustering the parameters based on their sensitivity indices and then classify them as significant or insignificant based on the clusters' mean value. Following ibid, we used k-means clustering with $k = 10$. The clustered parameters are given in Table 4.3 and the obtained mean index values are shown in Fig. 4.15 (c). As it is clear from this Figure the cluster, C_3, is the most sensitive one, which is followed by C_7, C_{10} and C_5. Note that C_3 contains the top three sensitive parameters, [X_{31}, X_{28}, X_{41}] (see Table 4.3). Cluster C_7 only contains X_{38}, which is the third sensitive parameter. Similarly, parameters [X_{37}, X_{42}, X_{76}] and [X_{32}, X_{47}, X_{48}, X_{91}] compose C_{10} and C_5, respectively. On the other hand, C_1 is the biggest yet the least sensitive of clusters. As we will show in the §4.3.4, this clustering of parameters also facilities analysing their impact on the design and comparing sensitive features obtained from FFD and PD.

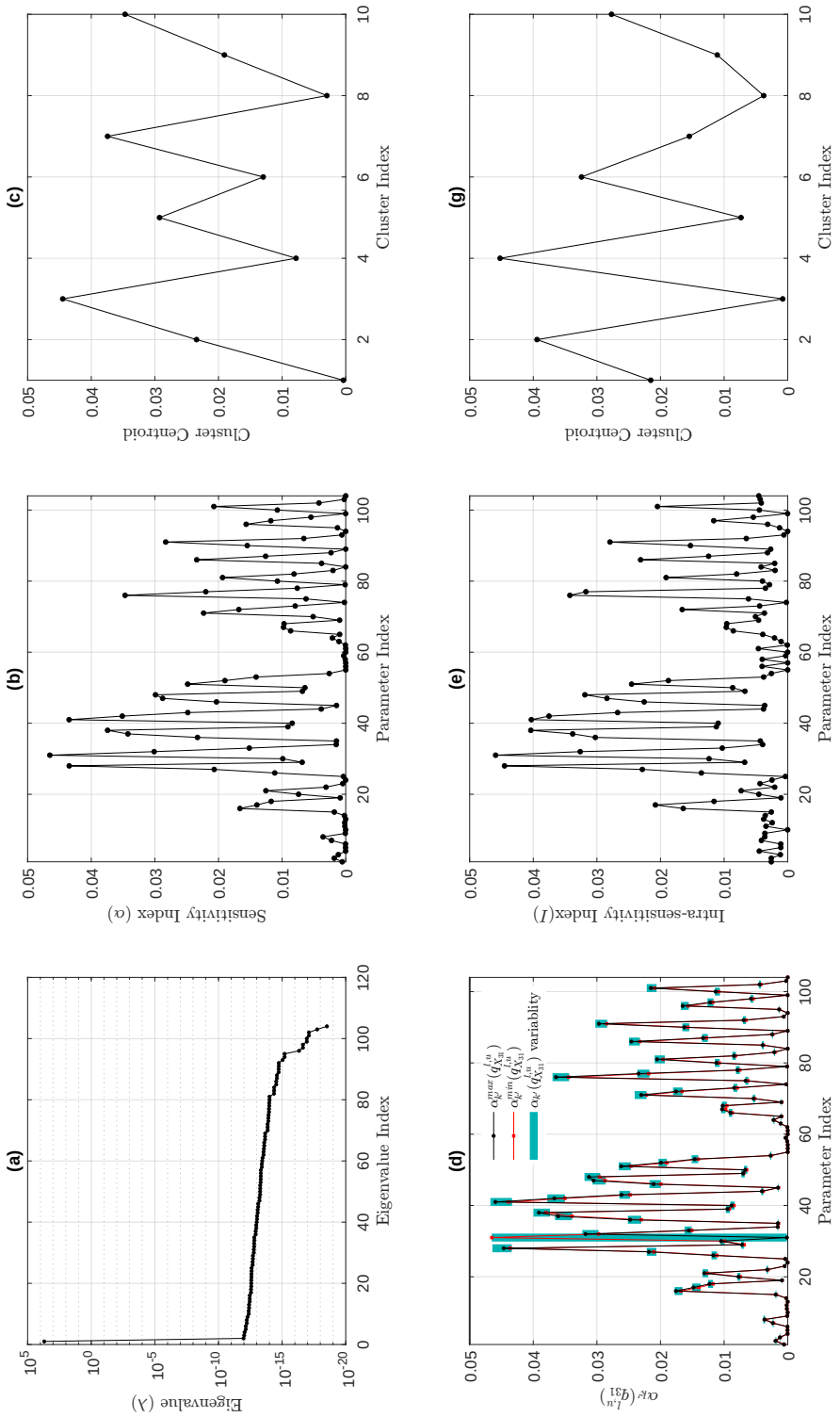


Figure 4.15: Sensitivity, intra-sensitivity and the k-means clustering results of all 104 FFD (Free-Form Deformation) parameters when ∇ (Volume of Displacement) is used as the QoI (Quantity of Interest): (a) eigenvalues, (b) sensitivity indices, (c) clusters' mean sensitivity indices of FFD parameters, (d) variational plot sensitivity plot of parameter X_{31}, (e) intra-sensitivity indices and (f) clusters' mean intra-sensitivity indices.

Table 4.3: Results of k-means clustering of all 104 parameters of FFD (Free-Form Deformation) based on their ∇-sensitivity indices (∇:=Volume of Displacement)

Clusters	Parameters
C_1	X_1, X_3, X_4, X_5, X_6, X_9, X_{10}, X_{11}, X_{12}, X_{13}, X_{14}, X_{19}, X_{23}, X_{24}, X_{25}, X_{34}, X_{35}, X_{45}, X_{55}, X_{56}, X_{57}, X_{58}, X_{59},X_{60}, X_{61}, X_{62}, X_{63}, X_{65}, X_{69}, X_{74}, X_{79}, X_{84}, X_{89}, X_{93}, X_{94}, X_{95}, X_{99}, X_{99}, X_{103}, X_{104}
C_2	X_{36}, X_{43}, X_{51}, X_{71}, X_{77}, X_{86}
C_3	X_{28}, X_{31}, X_{41}
C_4	X_{20}, X_{29}, X_{30}, X_{39}, X_{40}, X_{49}, X_{50}, X_{66}, X_{67}, X_{68}, X_{73}, X_{75}, X_{78}, X_{82}, X_{92}, X_{98}
C_5	X_{32}, X_{47}, X_{48}, X_{91}
C_6	X_{17}, X_{18}, X_{21}, X_{26}, X_{33}, X_{53}, X_{80}, X_{87}, X_{90}, X_{96}, X_{97}, X_{100}
C_7	X_{38}
C_8	X_2, X_7, X_8, X_{15}, X_{22}, X_{44}, X_{54}, X_{64}, X_{70}, X_{83}, X_{85}, X_{88}, X_{102}
C_9	X_{16}, X_{27}, X_{46}, X_{52}, X_{72}, X_{81}, X_{101}
C_{10}	X_{37}, X_{42}, X_{76}

Regional analysis and intra-sensitivity index of FFD parameters

Similar to PD, regional sensitivity analysis is performed with FFD to evaluate the intra-sensitivity of its parameters. Providing the 2D or 3D regional plots of 104 parameters is rather prohibitive so we are limiting ourselves to deliver the variation plot of the most significant parameters, X_{31} and the intra-sensitive results of all parameters; see Fig. 4.15 (d) and (e), respectively. Similar to PD, when the range of X_{31} is shrunk its sensitivity changes monotonically, which also induces variation in the sensitivity indices of another parameter as shown in Fig. 4.15 (d). Like PD, these variations are high in the sensitive parameters as compared to insensitive ones.

The intra-sensitivity is also analysed by clustering the parameters based on their intra-sensitivity indices. The clustering results are given in Table 4.4 and their mean values are shown in Fig. 4.15 (f). The cluster C_4 contains top two intra-sensitive parameters, X_{31} and X_{28} (see Table 4.4). C_2 is the second intra-sensitive cluster containing X_{38}, X_{41}

Table 4.4: Results of k-means clustering of all 104 parameters of FFD (Free-Form Deformation) based on their ∇-intra-sensitivity indices (∇:=Volume of Displacement)

Clusters	Parameters
C_1	X_{17}, X_{27}, X_{46}, X_{51}, X_{52}, X_{81}, X_{86}, X_{101}
C_2	X_{38}, X_{41}, X_{42}
C_3	X_3, X_5, X_6, X_{10}, X_{19}, X_{22}, X_{25}, X_{55}, X_{57}, X_{59}, X_{60}, X_{62}, X_{63}, X_{64}, X_{74}, X_{83}, X_{85}, X_{93}, X_{94}, X_{95}, X_{99}
C_4	X_{28}, X_{31}
C_5	X_{21}, X_{29}, X_{49}, X_{50}, X_{66}, X_{75}, X_{82}, X_{92}
C_6	X_{32}, X_{36}, X_{37}, X_{48}, X_{76}, X_{77}
C_7	X_{16}, X_{26}, X_{72}, X_{90}
C_8	X_1, X_2, X_4, X_7, X_8, X_9, X_{11}, X_{12}, X_{13}, X_{14}, X_{15}, X_{20}, X_{23}, X_{24}, X_{34}, X_{35}, X_{44}, X_{45}, X_{53}, X_{54}, X_{56}, X_{58}, X_{61},X_{65}, X_{69}, X_{70}, X_{71}, X_{73}, X_{78}, X_{79}, X_{80}, X_{84}, X_{88}, X_{89}, X_{96}, X_{98}, X_{100}, X_{102}, X_{103}, X_{104}
C_9	X_{18}, X_{30}, X_{33}, X_{39}, X_{40}, X_{67}, X_{68}, X_{87}, X_{97}
C_{10}	X_{43}, X_{47}, X_{91}

and X_{42}. The parameters X_{38} and X_{41} have similar intra-sensitive indices, giving both parameters third place in the intra-sensitivity ranking. It is noteworthy that in the case of FFD, the five most intra-sensitive parameters are also the most sensitive ones; however, their individual ranking is different. Furthermore, there are also few other parameters, such as X_{17} and X_{77}, which do not show significant sensitivity effect on ∇ but do possess some intra-sensitivity; X_{77}, is ranked 17^{th} and 10^{th} based on its sensitivity and intra-sensitivity, respectively.

4.3.4 Impact of sensitive and intra-sensitive parameters of FFD and PD on geometry

After identifying the sensitive and intra-sensitive parameters of PD and FFD, their impact on hull geometry is assessed to identify the geometrical features or regions, which are sensitive and intra-sensitive to ∇ and R_T. These results are then analysed for understanding whether these two approaches reveal similar sensitive and intra-sensitive features.

Fig. 4.16 (a) shows the plot of two-sided Hausdorff distance, $\mathcal{E}(X)$, calculated with Eq. (4.25) versus the parameter values of PD over their range. It can be observed that, apart from X_6, X_{14} and X_{17}, all parameters show maximum deviation from the baseline design at the upper limits of their range. For X_6, X_{14} and X_{17}, the largest geometric deviation occurs at the lower limits. Once the parametric values giving the maximum geometrical deviation is identified then it can be used in Eq. (4.24b) to provide the feature saliency maps like in Fig. 4.16 (b). Recall that these maps provide a graphical tool for the identification of affected regions of the hull when parameters change. It is also noteworthy that, apart from X_{14}, X_{15} and X_{17}, $\mathcal{E}(\mathbf{X})$ is zero at 0.5 as during the refinement of the design space their range remained unaffected and thus for these parameters parent hull stayed at the centre of the design space.

As it can be seen from Fig. 4.16 (b), X_9 and X_{15}, the two sensitive parameters for ∇ and R_T, affect the forward segment of the hull close to the bow. On the other hand, X_{10}, which is also a sensitive parameter for both ∇ and R_T influences the regions near the stern. Finally, the intra-sensitive parameter, X_6, affects the similar regions as of X_9 and X_{10}. From these results, it can be anticipated that these regions of the hull account for the most variation in ∇ and R_T. Moreover, R_T of the hull also significantly affected by X_7, which affects the shape of the bulbous bow and as explained earlier, it plays an important role in reducing R_T.

Similar to PD, after evaluating the clusters of sensitive and intra-sensitive parameters of FFD the hull features corresponding to these parameters are identified. In this case, the highest deviation in shape occurred when the value of all parameters is set at their upper limit. The first five images of Fig. 4.17(a) shows the feature saliency map of parameters in the two most sensitive and intra-sensitive clusters, which affect the forward segment of the hull. Whereas, the last two images are the feature saliency map of X_{76} and X_{91}, which are the only two parameters of third and fourth sensitive and intra-sensitive clusters affecting the aft segment of the hull. It can be seen that each of these parameters create a very localised effect on the surface. Moreover, regions identified by these parameters

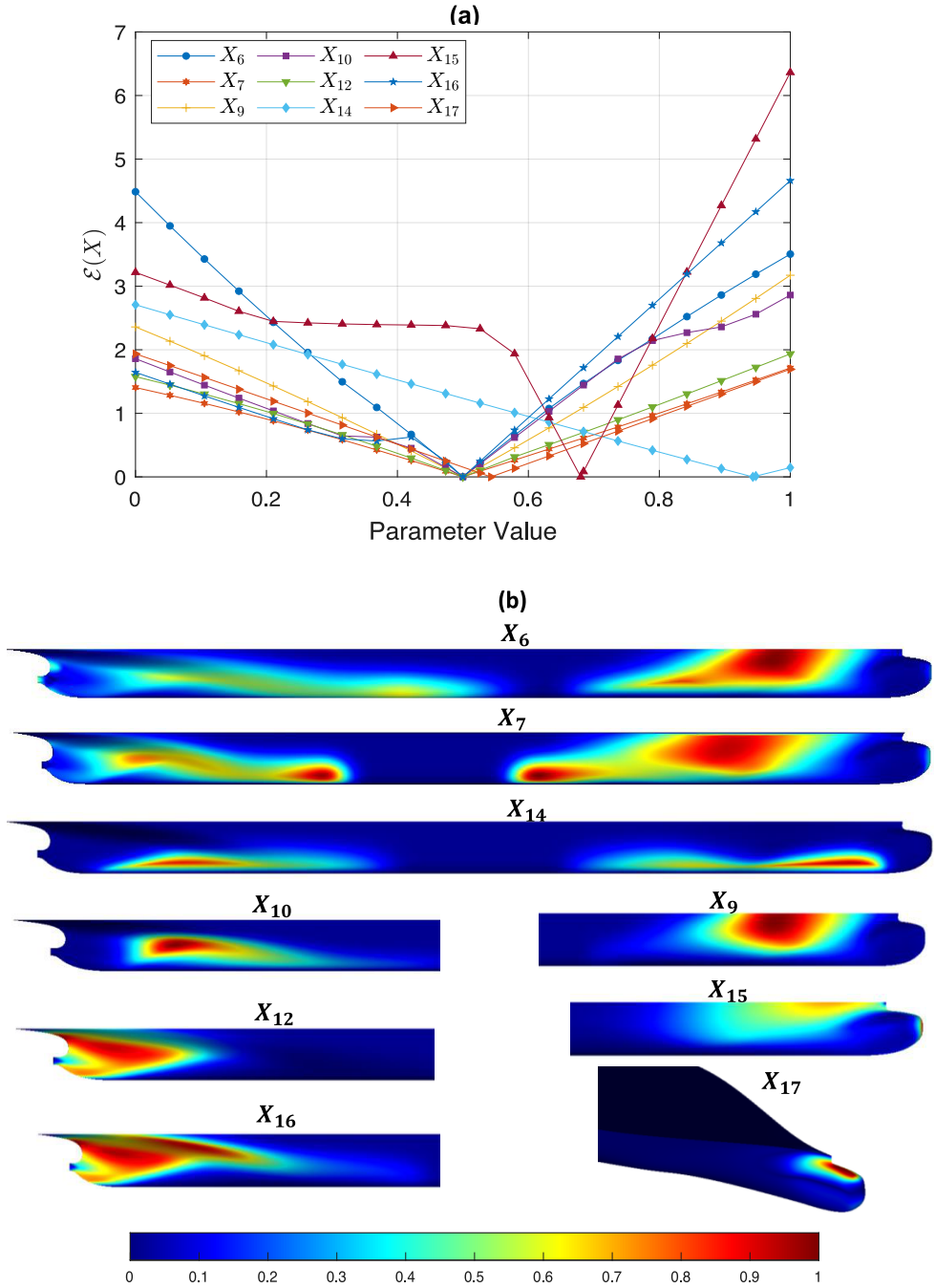


Figure 4.16: (a) Plot showing the two-sided Hausdorff distance ($\mathcal{E}(X)$) versus the PD (Procedural Deformation) parameter values over their entire range; (b) Feature saliency map of sensitive and intra-sensitive PD parameters of the parent ship hull.

do not correspond to the sensitive and intra-sensitive regions identified in the case of PD. Thus, a question could be raised here is that as the sensitivity and intra-sensitivity of FFD parameters is analysed in term of clusters so what effect it creates when the combined effect of these parameters is visualised on the hull surface. Therefore, we project the parameters in the first sensitive and intra-sensitive clusters on the hull surface, whose results are depicted in Fig. 4.17(b) and (c), respectively. Interestingly, the regions identified in this case show slight similarity with the sensitive regions identified by parameters X_9, X_{10} and X_{15} of PD. Consequently, it can be concluded that when the combined effect of FFD parameters is analysed then both parametrisation types, to some extent, can identify the similar type of features. However, the computational cost, for FFD is significantly high as it creates a high-dimensional design space.

4.3.5 Optimisation

In order to observe if sensitive and intra-sensitive parameters can result in the same optimal design when optimisation is performed in the original design space. We performed shape optimisation of the parent hull, parameterised with PD, in order to optimise its shape against R_T. In this optimisation process, three different design spaces are tested. The first design space is the original 24-dimensional with all hull parameters, while the second design space is of considerably lower dimension, constructed with the eight most sensitive parameters, namely X_7, X_9, X_{10}, X_{12}, X_{14}, X_{15}, X_{16} and X_{17}. Furthermore, the third design space, along with the eight most sensitive parameters, includes the most intra-sensitive parameter, namely X_6, which results in a 9-dimensional design space. The efficiency of these design spaces to obtained optimal design with faster convergence is analysed. For this optimisation, we utilised Jaya Algorithm (JA) [109], which is a newly proposed, simple yet effective metaheuristic 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 metaheuristic optimisation techniques, JA does not require

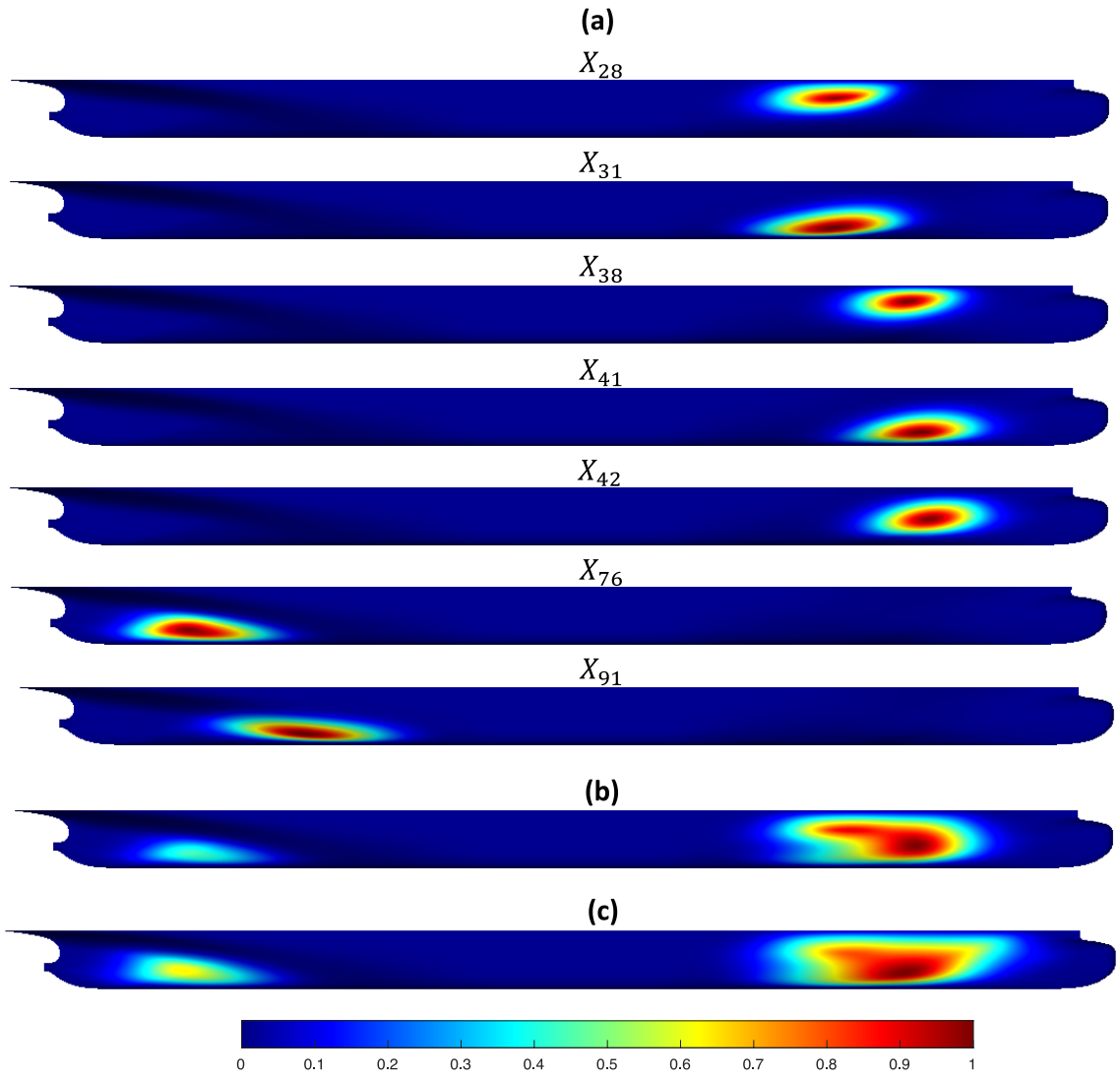


Figure 4.17: Feature-saliency map of (a) individual and (b, c) clustered sensitive and intra-sensitive FFD (Free-Form Deformation) parameters on parent hull form.

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. Moreover, as JA is a stochastic technique, which may provide different results in each run, we performed five different optimisation runs and in each run, a total of 50 iterations are conducted. Fig. 4.18 shows the average values of R_T in five runs. The optimal designs obtained from 24-

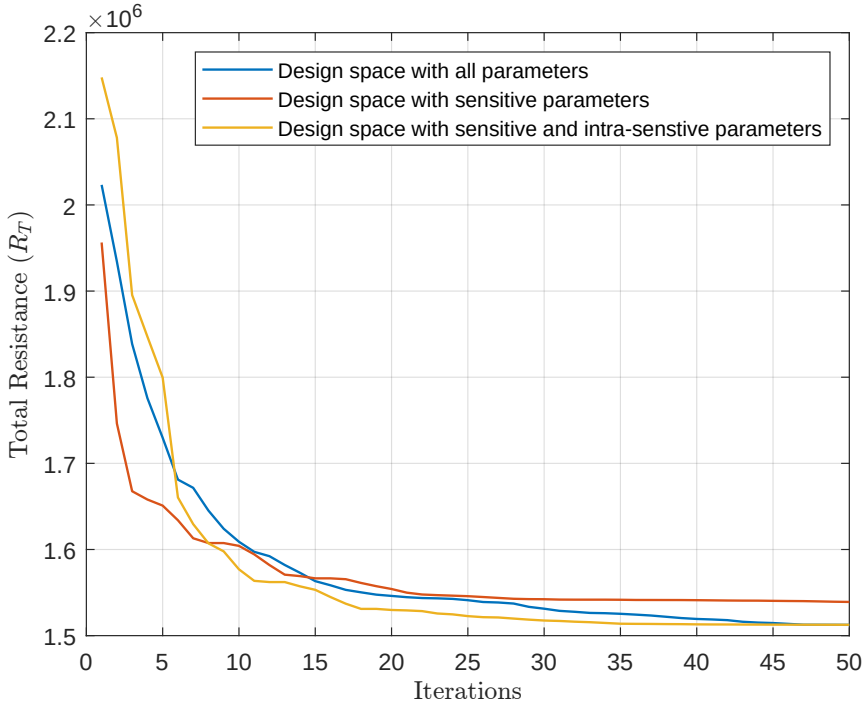


Figure 4.18: Results of optimisation of parent hull performed with PD when the design space is constructed using all parameters (blue curve), only sensitive parameters (red curve), and sensitive along with intra-sensitive parameters (yellow curve).

, 9- and 8-dimensional design spaces, which are shown in Fig. 4.19, have R_T equal to 1512518.018, 1539117.543 and 1512601.389 newtons, respectively. From these results, it can be seen that the final design obtained from the 24-dimensional space has slightly better performance than one obtained from the 8-dimensional design space, which is composed of only sensitive parameters. Moreover, it is interesting to note that when, along with sensitive parameters, intra-sensitive parameter, X_6, is included in the optimisation (i.e.,

9-dimensional design space) then the design obtained from this space has total resistance similar to the one obtained from the 24-dimensional space. Furthermore, the order of convergence of the optimiser in both lower-dimensional design spaces is faster. For instance, in the case of 8- and 9-dimensional design spaces, the convergence started even after the 20th iteration, whereas in the case of the 24-dimensional space, convergence is slower as the optimiser has to explore the high-dimensional design space and starts after the 30th iteration. Thus running optimisation in this space has a higher computational cost.

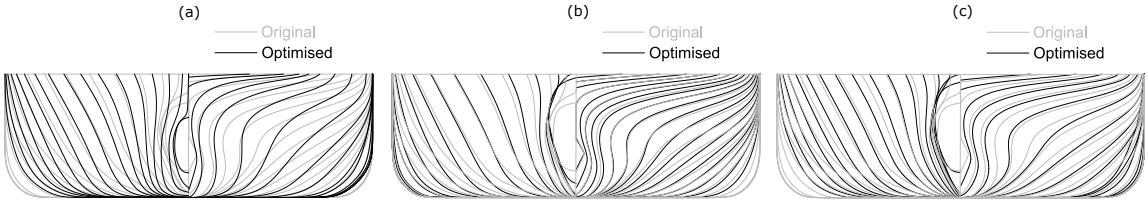


Figure 4.19: Comparison between the parent and optimised hulls obtained from (a) 24-, (b) 8- and (c) 9-dimensional design spaces.

4.3.6 Comparative studies

In this section we discuss the results of the comparative studies to verify the performance of components of the proposed approach. First, the convergence performance of Algorithm 1 is analysed when designs are sampled with DPS (Algorithm 2) or MC sampling. Secondly, the results of ASM and its proposed regional version are compared with the widely used Sobol's sensitivity and CSV (Contribution to the Sample Variance) techniques, respectively.

Comparison of DPS with MC: Fig. 4.20 shows the convergence plot of Algorithm 1 with MC over first 20 iterations in case of PD when ∇ is used as QoI. It can be seen that in comparison to DPS (see Fig. 4.8), the convergence of Algorithm 1 with MC is slow. The sensitivity indices and the parametric ranking are unstable not only for the insignificant parameters but also for the significant ones even after 20 iterations. This demonstrates that DPS outperforms MC in terms of achieving stable and robust results with respect to the least number of designs. Furthermore, we used box and whiskers plots, shown in Fig.

4.21, to check the variability of the sensitivity indices obtained with MC and DPS over 20 iterations. The results obtained from DPS have lower variability versus those obtained with MC, which again confirm the stable behaviour of Algorithm 1 when it is used in connection with DPS.

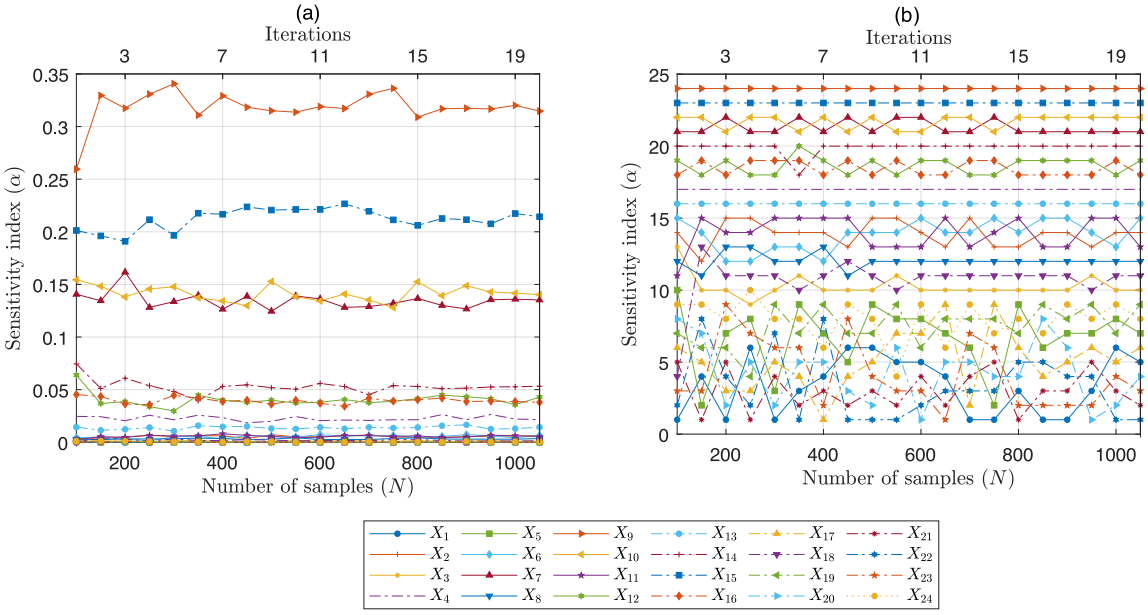


Figure 4.20: Plot showing the convergence of: (a) sensitive indices and (b) parametric ranking of PD parameters for ∇ (Volume of Displacement) versus the number of samples over the first 20 iterations of Algorithm 1 when it is used in conjunction with MC (Monte-Carlo).

Despite that, the computational cost of running DPS is higher versus to that of MC as it is based on the optimisation strategy. However, MC requires a large number of designs for convergence. As a consequence, the overall computational cost becomes much higher, especially for high-dimensional problems. The same conclusions can be drawn when this comparison is performed with R_T in place of ∇.

Comparison of ASM with Sobol's sensitivity analysis: Sensitivity results obtained with ASM (see Fig. 4.8 (a) and 4.21 (a)) are also compared with results obtained from Sobol's sensitivity analysis. At each iteration, $N = 100$ designs are sampled and

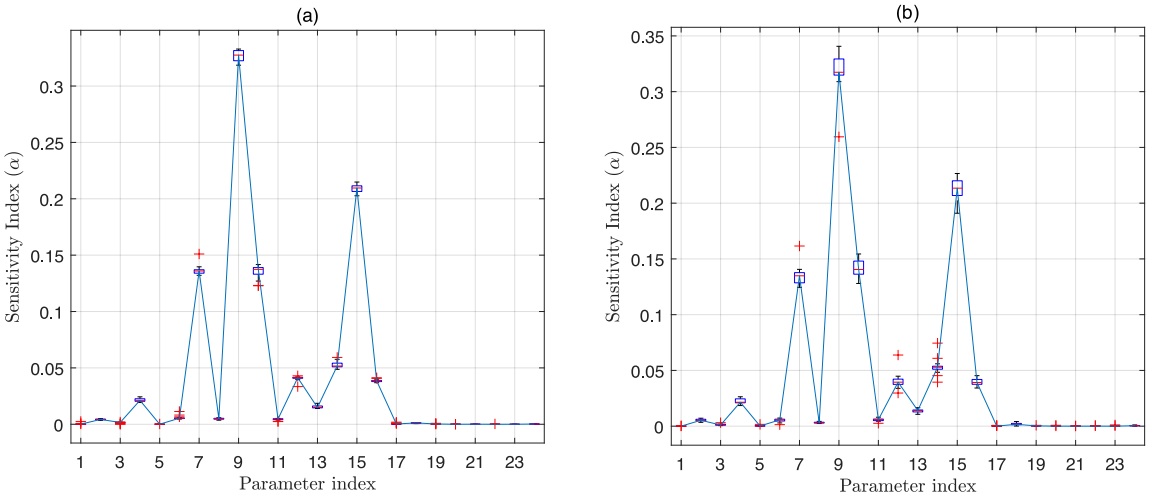


Figure 4.21: Box and whiskers plots of the sensitivity indices of the PD parameters for ∇ evaluated with Algorithm 1 when designs are sampled with: (a) the proposed DPS and (b) MC sampling.

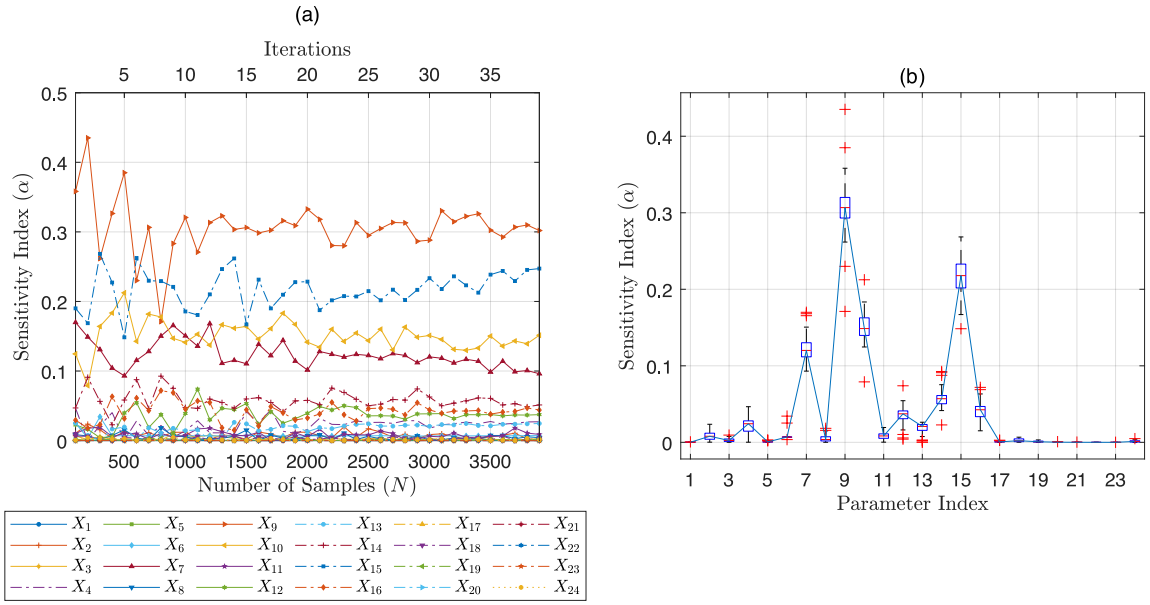


Figure 4.22: (a) Convergence and (b) box and whiskers plots of sensitivity indices of PD parameters for ∇ evaluated with Sobol's sensitivity analysis.

Sobol's analysis is performed until sensitivity indices stabilise. On the basis of [80], one may expect that both approaches behave similarly. However, our experiments show that Sobol's analysis exhibits slower convergence compared to ASM, as can be easily seen by comparing Figures 4.8 (a) and 4.22 (a). In Fig. 4.22 (a) sensitivity indices of all the parameters fluctuate highly up to ten iterations and tend to stabilise only after 20 iterations, where $N = 2000$. The variation of these sensitivity indices is given in Fig. 4.22 (b), which in comparison to the ASM results in Fig. 4.21 (a), show higher variability even when they are computed with a large number of samples.

Comparison of ASM-based regional analysis with CSV: We also compare the proposed regional approach with CSV to investigate if the latter can help to identify the intra-sensitivity of parameters. CSV results of PD with ∇ for the sensitive parameters are shown in Fig. 4.23. It can be seen that CSV only informs how ∇ varies over the quantile range of a parameter and does not provide any information on how the perturbation in the range of a parameter influences the sensitivity of other parameters. It does, however, identifies the regions of the range of any parameter where ∇ have the highest variability compared to other regions. For instance, X_9 plot in Fig. 4.23 shows that ∇ has higher variability over $[0, 0.6]$ than over $[0.6, 1]$. If the CSV curve (highlighted in red) close to the diagonal then the variability of QoI is the same over the entire range of the parameter, such behaviour is represented by X_{16} and also somewhat by X_6, which from our analysis resulted as the most intra-sensitive parameter.

4.4 Conclusions and future works

In the previous sections, we presented a methodology for investigating regional non-uniformities of parametric sensitivity for facilitating the creation of viable design spaces for free-form shape optimisation. The proposed regional analysis is based on ASM (Active Subspace Method), which not only identifies regions of high parametric sensitivity against the chosen QoI's (Quantity of Interest) but also reveals the so-called intra-sensitive parameters

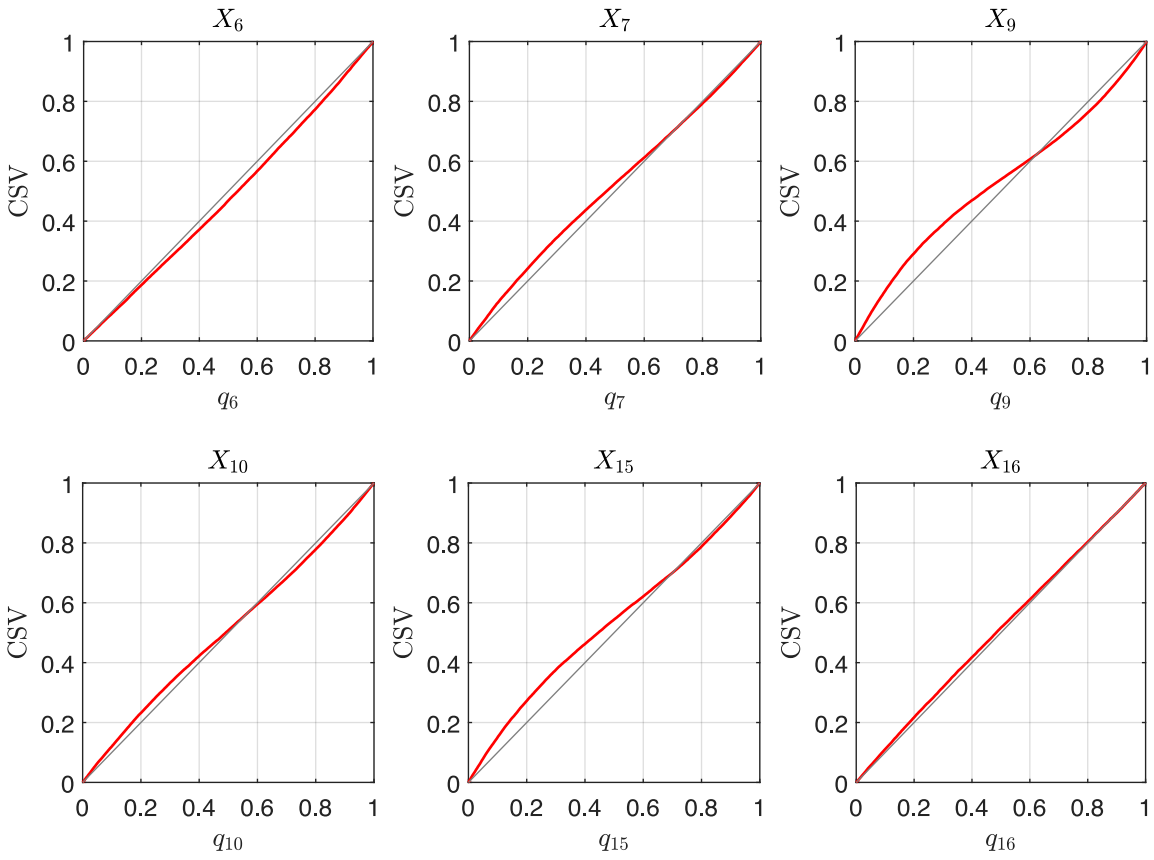


Figure 4.23: Contribution to the sample variance plots of the most sensitive parameters of PD showing the variability of ∇ over the parametric range.

whose perturbation in the range has a strong impact on the sensitivity of the remaining parameters. Once an intra-sensitive parameter is identified its range can be refined to obtain the desired sensitivity ranking of the parameters or to reduce uncertainty in the sensitivity results.

Shape optimisation in the context of contemporary industrial engineering applications usually involves high-dimensional design spaces, which requires a large number of samples for stable and robust results. Therefore, an iterative approach is utilised to evaluate the sensitivity and intra-sensitivity of the parameters, which progressively increases the number of samples during the iterations. At each iteration, designs are sampled with a Dynamic Propagation Sampling approach. To obtain uniformly distributed samples, our approach utilises the criteria of space-filling and non-collapsing along with the so-called repulsive criterion for improving diversity by taking on board designs from previously unexplored regions of the design space. This iterative process is terminated when the ranking of the sensitive parameters becomes stable. Once sensitive and intra-sensitive parameters are secured then free-form features corresponding to these parameters are evaluated using a feature saliency map. This map, generated via a Hausdorff distance-based approach, provides a visual tool to detect features or regions of the design affected by these parameters.

To verify the performance of the proposed pipeline a container ship hull is used, parametrised with two different parametric modellers based on procedural and free-form deformation. The sensitivity and intra-sensitivity of their parameters are evaluated against two QoI's, namely the volume of displacement and the total resistance, which are used in all phases of ship design. Finally, comparative studies presented with regard to the performance of dynamic propagation sampling, ASM-based sensitivity and regional analysis showed that these components of the proposed scheme result in faster convergence and more robust and reliable results.

Our medium-term plans include the exploitation of intra-sensitivity in shape optimisation problems in the area of energy-saving devices, e.g., bulbous-bow modifications, pro-

peller fin attachments, with QoI's involving the use of the medium- and high-fidelity modern hydrodynamic solvers, such as IGA[4]-based Boundary Element Methods [110, 111], and RANS[5]-based CFD solvers [112], respectively. We are also keen to work on the usage of shape integral properties as QoI to infer the sensitivity of parameters for design's physics.

[4]IGA: IsoGeometric Analysis
[5]RANS: Reynolds Averaged Navier Stokes

Chapter 5

Geometric moment-dependent global sensitivity analysis without simulation data

5.1 Introduction

In this chapter, we aim to address the aforementioned challenges associated with SA by offloading the evaluation of parametric sensitivities from physical quantities to relatively inexpensive quantities compared to physical ones but provide important clues about the form distribution and validity of the design. More specifically, it is well known that shape's integral properties, such as geometric moments and their invariants [14,51] serve as a geometric foundation for different designs' physical analyses. Like physics, they rely strongly on design's geometry, but their evaluation is substantially less expensive. Therefore, we propose a geometric moment-dependent SA approach that harnesses the geometric variation of designs in a design space using geometric moments as a geometrical Quantity of Interest (QoI) to measure parametric sensitivities. These results can serve as a prior estimation of parametric sensitivities and use to construct a design space of lower dimen-

sion with only a subset of highly/strongly sensitive parameters for shape optimisation performed against physical QoI. This approach can significantly reduce the computational time because, typically, sensitivities are learnt directly with physical QoI, which can add a heavy computational burden on the entire design process as one has to perform computationally intensive physical simulations for both SA and shape optimisation. Fig. 5.1 shows the systematic workflow of the proposed approach, which uses geometric moments for SA and design's physics for shape optimisation. The selection of geometric moments for SA in our work is motivated by the following fundamental insights:

1. Geometric moments of a shape are intrinsic properties of its underlying geometry and act as a unifying medium between geometry and its physical evaluation [113, 114].

2. Physical analysis requires the evaluation of such integral properties of the geometry such as the stiffness and mass matrices, and moments of a domain are sufficient to ensure accurate integration of a large class of integrands [14, 115].

3. Like physics, geometric moments also act as a compact shape signature or descriptor to a specific design falling in a specific category, which facilitates various shape processing tasks [116–118].

In this work, we show through extensive experiments the competitive performance of the geometric moments for making an informed decision on the sensitivity of parameters without performing computationally intensive physical simulations. The results of SA via geometric moments permit to categorised a prior the design parameters as *strongly sensitive*, *moderately sensitive*, *weakly sensitive*, and *insensitive*. According to Sheikholeslami et al. [37] and Klepper [119], when such categorisation of parameters is available, then computationally efficient SA with physics can be performed for each category. As explained earlier in this section, different physical analyses have a dependence on shape integral properties such as geometric moments, but during any design process, there are many physical criteria (varying from application to application) that have to be investigated and may not

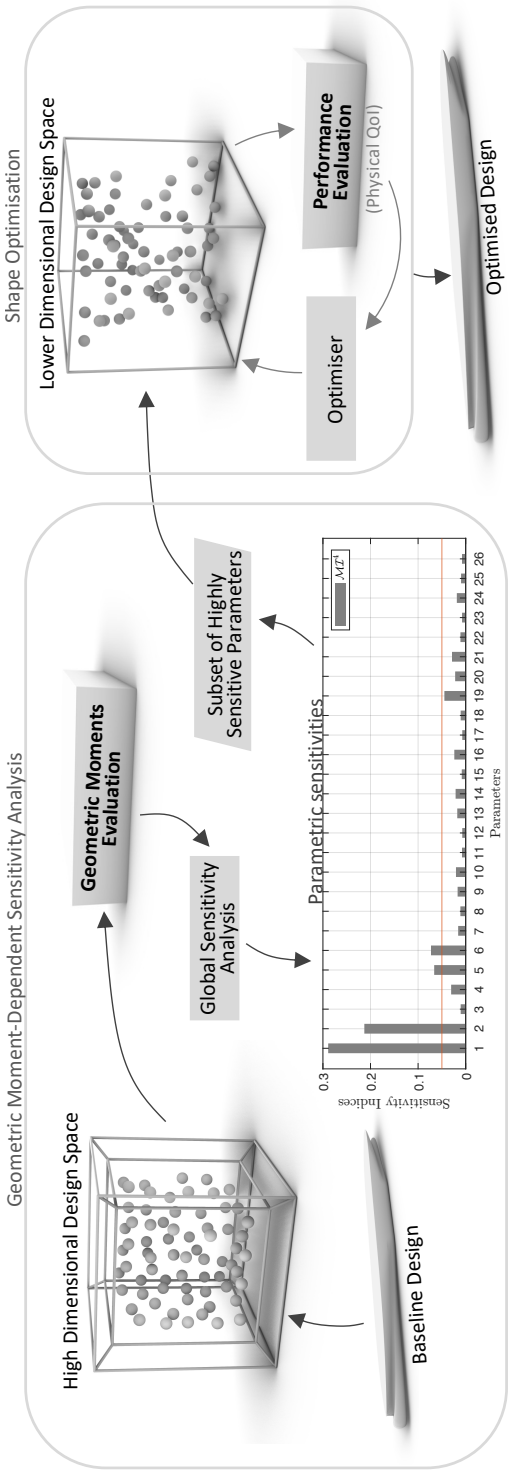


Figure 5.1: Workflow of the proposed approach, which uses a shape-signature vector containing geometric moments of variant order for sensitivity analysis and design's physics for shape optimisation.

be dependent on geometric moments. *Thus, the use of moments does not aim to eliminate the need to perform SA with respect to physics but rather to support the designer for a prior check regarding the sensitivity of parameters for those physical quantities that are computationally expensive and share relevance with geometric moments.* Therefore, we restrict our aim to exploring the capability of these geometric moments in the context of ship-hull design, namely, with regard to their capability to reveal the sensitivity of its parameters for the wave-making resistance coefficient (C_w), which is one of the significant components of total ship's resistance and a critical design criterion.

Wave-making is caused when an object moves on or near the free surface of the water. The waves are generated from the variation of pressure over the wetted surface of the ship and carried behind it in the form of the so-called Kelvin wave pattern through a mechanism that is due to the kinematic and dynamic conditions satisfied on the free surface of the ocean. The importance of C_w reduction at the preliminary design stage, its effect on hull geometry and its connection with geometric moments will be discussed and formulated in §5.3.1. To experimentally demonstrate the effectiveness of geometric moments, we used two ship hulls parameterised with 26 and 27 parameters using two different techniques based on *Procedural Deformation* (PD) [1] and *Global Modification Function* (GMF) [12], respectively. The former parameterises a hull geometry constructed using the NURBS (non-uniform rational B-splines) representation and parameters directly associated with the hull's key features. In contrast, the latter is defined directly on a design grid and creates a global surface deformation. For the hull parameterised with PD, its C_w is evaluated using an inviscid BEM (Boundary Element Method) isogeometric solver [15]. For the GMF-based hull, C_w is obtained with a method based on the linear potential flow theory as well [120]. Geometric moments for both hulls are evaluated via the divergence theorem [121], which is applied on the triangulated mesh surfaces of the hull.

To commence SA, we first construct the so-called *Shape-Signature Vector* (SSV), which acts as a unique descriptor for the shape and contains all the geometric moments up to a certain order. To better correspond to C_w, all the moments in this vector are formulated

to be invariant to translation and scaling. A global variance-based SA [78] is performed concerning SSV and C_w. Here, the former is purely a vector quantity containing the moment of various orders, while the latter is a scalar one but computationally expensive to evaluate. Therefore, learning sensitivity to SSV requires implementing a multivariate output SA technique, such as covariance decomposition [122], which provides generalised sensitivity indices of design parameters to all moments in SSV. Afterwards, a series of experimentations are performed to identify a common set of sensitive parameters between SSV and C_w. Furthermore, the higher the order of SSV is, the better it can describe the shape, and as a result, its parametric sensitivity better correlates with C_w. However, evaluation of higher-order moments can be prone to numerical noise. Therefore, following various other applications of moments in literature [113, 123] we restrict our analysis to geometric moments up to fourth-order. We also study the local effect of geometric moments evaluated after segmenting the hull shape to compensate for this. To further demonstrate the effectiveness of geometric moments, for each test case, two shape optimisations are performed in the design spaces constructed with parameters sensitive to SSV and C_w. Their results are compared to optimal design obtained when optimisation is performed within the actual high-dimensional design space.

5.1.1 Geometric moments in design and analysis

From a geometric point of view, these moments are typically used to evaluate the overall volume enclosed by the shape, its centre of mass, and moment of inertia. Geometric moments are used over a wide spectrum of applications ranging from probability and statistics to signal processing, computing tomography [124], object recognition [118], shape retrieval [117], rigid body transformation [116], feature extraction [53] etc. In physical analyses, they are used for governing equation of motions for flows around a body [113], integrating accurately implicit functions [115], modelling nonlinear material properties in the cut-cell method [125], simplifying history-dependent material modelling in the case of crack propagation [126], and material field modelling to develop an integral representation

for fields that supports a wide range heterogeneous data [114]. Recently, meshfree methods have also been developed [14, 127, 128], which use moments to generate quadrature rules for the geometric domain to aid the interoperability between CAD representation and its physics.

Similar to the present work, Taber et al. [14] used a moment-vector for composing components of moments of a different order; however, there is a slight difference in their construction and the number of moments they contain. A new integration technique called Shape Aware Quadratures (SAQ) was proposed by Vaidyanathan, and Vadim [127] to efficiently integrate arbitrary integrable functions over arbitrary 2D/3D domains even in the presence of small features. It uses different derivative-based shape sensitivities of first and second-order to construct shape correction factors used in the moment-fitting equations. These shape correction factors ensure that the quadrature rule determined by the moment-fitting equations is aware of the shape of the integration domain, especially associated with the small features. Christian et al. [128] proposed a new type of analysis pipeline, eXtended Finite Element Method (XFEM), mainly to support smooth interoperability between CAD and physical simulation during shape optimisation. The core of their contribution lies in using a moment-fitting technique to compute on-the-fly a modified set of quadrature rules that accurately handle integration over curved domains of varying shape and size, bounded by NURBS and planar patches, and evaluate shape derivatives with respect to these rules to quantify the shape sensitivities caused on the volume integral with the change of the design parameters. The derivative-based shape sensitivities evaluated in [127,128] are usually referred to as local sensitivities [31, 75], which enables identifying the local influence of a single parameter on the QoI. As explained earlier, in this work, we use GSA, which provides a holistic view of the influence of all design parameters on the QoI in question.

The remainder of this chapter is organised as follows: Section 5.2 discusses the problem formulation, evaluation of geometric moments and SA for problems with univariate and multivariate outputs. A detailed discussion on the relevance of geometric moments with C_w, along with the description of the test cases, is given in Section 5.3. The numerical

results of the proposed technique are provided in Section 5.4. Concluding remarks and plans for future work are presented in Section 5.5.

5.2 Geometric moment-dependent sensitivity analysis

This section provides an in-depth description of the proposed approach, including the general assumptions, mathematical formulation of geometric moments and their invariants and a brief overview of SA for univariate and multivariate outputs models.

5.2.1 Problem formulation

Let a geometric design $\mathcal{G}$ be parameterised with a set of n continuous design parameters $\mathbf{t} = \{t_i, i = 1, 2, \ldots, n\} \in \mathcal{X} \subseteq \mathbb{R}^n$. Here $\mathcal{X}$ is the n-dimensional solution/design space, bounded by lower $\mathbf{t}^l$ and upper $\mathbf{t}^u$ limits of the associated parameters (i.e., $\mathcal{X} := \{t_i^l \leq t_i \leq t_i^u, \forall i \in \{1, 2, \ldots n\}\}$). Moreover, all the elements of $\mathbf{t}$ are assumed to be statistically independent from each other, i.e., $p_{\mathbf{t}}(\mathbf{t}) = \prod_{i=1}^{n} p_{t_i}(t_i)$, where $p_{\mathbf{t}}(\mathbf{t}) : \mathbb{R}^n \to \mathbb{R}$ represents the Probability Density Function (PDF) of $\mathbf{t}$ and $p_{t_i}(t_i)$ is the marginal PDF of t_i. Now, the objective of the present work is to assess the sensitivity indices, $\mathbf{SI} = \{SI_1, SI_2, \ldots, SI_n\}$, of each element of $\mathbf{t}$ with respect to geometrical QoI, such as geometric moments of $\mathcal{G}$. Therefore, we assume to posse a shape-signature vector, $\mathcal{MI}^s$, which contains all the geometric moments from 0^{th} to s^{th} order. Construction of $\mathcal{MI}^s$ will be discussed in detail in the subsequent subsections. Once the SA is performed, the aim is to find a subset $\mathbf{t}_{\mathcal{MI}}$ of m highly sensitive parameters whose sensitivity index is greater than a threshold, ϵ, where m is favourable to be less than n. The subset of m parameters forms a design space $\mathcal{X}_{\mathcal{MI}}$ of reduced dimension, which is exploited to expedite the shape optimisation carried out directly with a physical QoI; represented as $y = g(\mathbf{t}) : \ \mathcal{X} \subseteq \mathbb{R}^n \to \mathbb{R}$. In conclusion, the overall problem can be stated as follows:

Sensivity:

$$\begin{aligned} \text{Find } \mathbf{t}_{\mathcal{MI}} \subseteq \mathbf{t} \quad & \text{sensitive w.r.t. } \mathcal{MI}^s \\ \text{where} \quad & \mathbf{t} \in \mathcal{X} \subseteq \mathbb{R}^n \\ \text{Construct } \mathcal{X}_{\mathcal{MI}} \quad & \text{such that} \\ & \mathbf{t}_{\mathcal{MI}} \in \mathcal{X}_{\mathcal{MI}} \subseteq \mathbb{R}^m \\ & m < n \end{aligned} \tag{5.1}$$

Optimisation:

$$\begin{aligned} \text{Find } \mathbf{t}^*_{\mathcal{MI}} \in \mathbb{R}^m \quad & \text{such that} \\ g(\mathbf{t}^*_{\mathcal{MI}}) = & \min_{\mathbf{t}_{\mathcal{MI}} \in \mathcal{X}_{\mathcal{MI}}} g(\mathbf{t}^*_{\mathcal{MI}}) \end{aligned}$$

5.2.2 Geometric moments

We shall use moments for quantifying the shape of an object (design, in the so-far terminology) $\mathcal{G}$ of finite extent, defined by the following formula:

$$M^{p,q,r}(\mathcal{G}) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} x^p \; y^q \; z^r \; \rho(x,y,z) \; \mathrm{d}x\mathrm{d}y\mathrm{d}z, \quad p,q,r \in \{0,1,2,\dots\}, \tag{5.2}$$

which gives the $s^{th}-$order geometric moment of $\mathcal{G}$, where $s = p+q+r$ and $\rho(x,y,z) = 1/0$ for $(x,y,z) \in / \notin \mathcal{G}$, respectively. Given a non-negative integer s, the vector $\mathbf{M}^s$ will contain which contains $(s+1)(s+2)/2$ moments $M^{p,q,r}(\mathcal{G})$ such that $p+q+r = s$. The ideal order of $\mathbf{M}$ will result in a vector containing geometric moments capturing not only global features of the shape but also the local features. For instance, moment of $(s=2)^{th}-$order contains

$$\mathbf{M}^2 = \begin{bmatrix} M^{2,0,0}(\mathcal{G}) & M^{0,2,0}(\mathcal{G}) & M^{0,0,2}(\mathcal{G}) & M^{1,1,0}(\mathcal{G}) & M^{1,0,1}(\mathcal{G}) & M^{0,1,1}(\mathcal{G}) \end{bmatrix}. \tag{5.3}$$

As pointed out in [14], moments can be thought of as projections (with respect to L^2 inner product) of ρ onto any polynomial basis, such as monomials, Legendre polynomials, etc. In Mathematical Analysis, the classical moment problem, which has been treated by various famous mathematicians such as Markov in 1883 and Stieltjes in his famous 1894 chapter on: "Recherchers sur les fractions continues", can be simply stated as follows: *Recover a function* $f(x)$ *given its moments* $M^p = \int x^p f(x)dx, p = 0, 1, ...$ In all these guises, the moment problem is recognised as a notoriously difficult inverse problem, often leading to the solution of very ill-posed systems of equations that usually do not have a unique solution [124].

In Eq. (5.2), if $\rho(x, y, z)$ represents the volume density then the zero- and first-order moments, $M^{0,0,0}(\mathcal{G})$, $M^{1,0,0}(\mathcal{G})$, $M^{0,1,0}(\mathcal{G})$, and $M^{0,0,1}(\mathcal{G})$, are widely used in computer graphics, CAD and engineering for computing the object volume, $\mathcal{V} = M^{0,0,0}(\mathcal{G})$, and the coordinates of the centre-of-volume:

$$\mathbf{c}(\mathcal{G}) = \begin{bmatrix} C_x \\ C_y \\ C_z \end{bmatrix} = \begin{bmatrix} \frac{M^{1,0,0}(\mathcal{G})}{M^{0,0,0}(\mathcal{G})} \\ \frac{M^{0,1,0}(\mathcal{G})}{M^{0,0,0}(\mathcal{G})} \\ \frac{M^{0,0,1}(\mathcal{G})}{M^{0,0,0}(\mathcal{G})} \end{bmatrix}. \tag{5.4}$$

If $\rho(x, y, z)$ is the PDF of a continuous random variable, then $\mathbf{M}^0$, $\mathbf{M}^1$, $\mathbf{M}^2$, $\mathbf{M}^3$ and $\mathbf{M}^4$, represent the total density, mean, variance, skewness and kurtosis of the random variable, respectively. Moreover, the moments of second-order can be organised in a second rank tensor, the moment of inertia tensor (MoI), which is represented as follows:

$$\text{MoI} = \begin{bmatrix} M^{0,2,0}(\mathcal{G}) + M^{0,0,2}(\mathcal{G}) & -M^{1,1,0}(\mathcal{G}) & -M^{1,0,1}(\mathcal{G}) \\ -M^{1,1,0}(\mathcal{G}) & M^{2,0,0}(\mathcal{G}) + M^{0,0,2}(\mathcal{G}) & -M^{0,1,1}(\mathcal{G}) \\ -M^{1,0,1}(\mathcal{G}) & -M^{0,1,1}(\mathcal{G}) & M^{2,0,0}(\mathcal{G}) + M^{0,2,0}(\mathcal{G}) \end{bmatrix}. \tag{5.5}$$

An appropriate combination of geometry and its moments results in a vector that better captures the shape's intrinsic features and offers a more accurate and unique shape representation that acts as its descriptor or signature [116]. The more moments we use, the better we capture the shape's intrinsic features. Our moment-driven SSV is represented by $\mathcal{M}^s = \left[\mathbf{M}^0, \mathbf{M}^1, \mathbf{M}^2, \ldots, \mathbf{M}^s\right]$, where s is appropriately large to cover the shapes of interest [116]. Theoretically, s ranges from 0 to ∞, though there exist classes of objects for which s is finite when, e.g., dealing with the class of the so-called quadrature domains in the complex plane [129] or when approximating convex bodies using Legendre moments [130]. The geometric moments of Γ can be thought of as projections (with respect to L^2 inner product) of ρ onto any polynomial basis, such as monomials, Legendre polynomials, etc. [14]. In Mathematical Analysis, the classical moment problem, which has been treated by various famous mathematicians such as Markov in 1883 and Stieltjes in his famous 1894 chapter on: "Recherchers sur les fractions continues", can be stated as follows: *Recover a function $f(x)$ given its moments $M^p = \int x^p f(x)dx, p = 0, 1, ...$* In all these guises, the moment problem is recognised as a notoriously difficult inverse problem, often leading to the solution of very ill-posed systems of equations that usually do not have a unique solution [124].

There exists a variety of methods available in the literature for computing geometric moments, which use either lower-order approximating mesh [131] or high-order surface [51] representations, such as B-splines and NURBS, of $\mathcal{G}$. The most commonly used method is Gauss's divergence theorem [121], which evaluates geometric moments by converting volume integrals to integrals over the surface bounding the volume. In 5.2.2, we summarise the evaluation of geometric moments using the divergence theorem for a triangulation $S = \bigcup_{i=1}^{N} T_i$ approximating the surface bounding $\mathcal{G}$, where N is the total of triangular elements T.

Geometric moment invariants

The moments in $\mathbf{M}^s$ are variant with respect to rigid and non-rigid transformations, such as translation, rotation and scaling [132]. However, most physical quantities are invariant to either all or some of these transformations. For instance, evaluating C_w for the ship is invariant to translation and scaling if assessed at a certain Froude number. Therefore, to measure the sensitivity of these parameters with respect to the geometry, the invariant of these geometric moments with respect to translation and scaling has to be secured. A description of geometric moment invariants with respect to translation and scaling presented in this section and their other invariants can be found in [132].

If Eq. (5.2) is applied for $\mathcal{G}$, while placing it at its centiod, $\mathbf{c}(\mathcal{G}) = (C_x, C_y, C_z)$, then we get the so called *central geometric moment* of $s^{th}-$order, which is invariant to translation and is expressed as:

$$\mu^{p,q,r}(\mathcal{G}) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}(x-C_x)^p\ (y-C_y)^q\ (z-C_z)^r\ \rho(x,y,z)\ \mathrm{d}x\mathrm{d}y\mathrm{d}z. \tag{5.6}$$

It is noteworthy that as $\mathcal{G}$ is placed at its centroid; therefore the first-order moment is zero, i.e., $[\mu^{1,0,0}, \mu^{0,1,0}, \mu^{0,0,1}] = 0$. To achieve invariance of $\mu^{p,q,r}$ to scaling we assume that $\mathcal{G}$ is uniformly scaled by a factor λ, which gives

$$\hat{\mu}^{p,q,r}(\hat{\mathcal{G}}) = \lambda^{p+q+r+3}\mu^{p,q,r}(\mathcal{G}). \tag{5.7}$$

Then, one can easily conclude that

$$MI^{p,q,r} = \frac{\mu^{p,q,r}}{(\mu^{0,0,0})^{1+(p+q+r)/3}} \tag{5.8}$$

is an invariant moment form for $\mathcal{G}$ under uniform scaling and translation [132]. For any non-negative integer, s, the moment invariant vector, $\mathbf{MI}^s$ contains all the moments invariant to translation and scaling such that $p+q+r = s$. By definition this invariance

satisfies $MI^{0,0,0} = 1$ and $\mathbf{MI}^1 = \left[MI^{1,0,0}, MI^{0,1,0}, MI^{0,0,1}, MI^{1,1,0}, MI^{1,0,1}, MI^{0,1,1}\right] = 0$. As $M^{0,0,0}$ represents the volume of $\mathcal{G}$, which is intrinsically invariant to translation; therefore, the invariant SSV, $\mathcal{MI}^s$, contains $M^{0,0,0}$ instead of $MI^{0,0,0}$ (i.e., $\mathcal{MI}^s = \left[M^{0,0,0}, \mathbf{MI}^2, \mathbf{MI}^3, \ldots, \mathbf{MI}^s\right]$).

Moment computation

To start the computation of geometric moments of $\mathcal{G}$, let be given a vector field $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^3$ over $\mathcal{V}$, whose boundary is piece-wise smooth surfaces. The divergence theorem states that the volume integral of the divergence (div) of $\mathbf{f}$ over $\mathcal{V}$ equals the surface integral of the normal component $\hat{\mathbf{n}}$ of $\mathbf{f}$ over triangulation S, which can be formalised as

$$\operatorname{div}(\mathbf{f}) = \sum \frac{\partial f_i}{\partial t_i} \tag{5.9}$$

$$\int_V \operatorname{div}(\mathbf{f}) \, \mathrm{dV} = \int_S \mathbf{f} \cdot \hat{\mathbf{n}} \, \mathrm{d}S. \tag{5.10}$$

With Eq. (5.10) we convert the volume integrals, which are difficult to evaluate, into surface integrals that are easy to evaluate over S. However, this theorem is only applicable if $\mathbf{f}$ is continuous and have continuous first partial derivatives in the region containing $\mathcal{V}$. To evaluate moments using this theorem, consider the following field:

$$\mathbf{f} = \frac{1}{3} x^p \; y^q \; z^r \left(\frac{x}{p+1}\hat{\mathbf{i}} + \frac{y}{q+1}\hat{\mathbf{j}} + \frac{z}{r+1}\hat{\mathbf{k}} \right). \tag{5.11}$$

and thus

$$M^{p,q,r}(\mathcal{G}) = \int_V \operatorname{div}(\mathbf{f}) \, \mathrm{d}V = \sum_{i=1}^{N} \int_{T_i} \mathbf{f} \cdot \hat{\mathbf{n}} \, \mathrm{d}S_i, \tag{5.12}$$

where $\hat{\mathbf{n}}_i$ is the unit normal vector on the triangle T_i, which can be represented as a linear parametric surface as

$$S_i(u,v) = \boldsymbol{\alpha}_i u + \boldsymbol{\beta}_i v + \mathbf{c}_i, \quad (u,v) \in \Omega_i \subset \mathbb{R}^2, \tag{5.13}$$

here Ω_i can be taken to be the triangle with vertices $(0,0)$, $(1,0)$, $(0,1)$. Then

$$M^{p,q,r}(\mathcal{G}) = \sum_{i=1}^{N} \int_{T_i} \mathbf{f} \cdot \hat{\mathbf{n}}_i \sqrt{\mathbf{E}_i \mathbf{G}_i - \mathbf{F}_i^2}\, \mathrm{d}u\mathrm{d}v, \tag{5.14}$$

where

$$\mathbf{E}_i = S_{i,u} \cdot S_{i,u} = |\boldsymbol{\alpha}_i|^2, \quad \mathbf{F}_i = S_{i,u} \cdot S_{i,v} = 0, \quad \mathbf{G}_i = S_{i,v} \cdot S_{i,v} = |\boldsymbol{\beta}_i|^2. \tag{5.15}$$

Here, $\mathbf{E}_i$, $\mathbf{F}_i$ and $\mathbf{G}_i$ are the constant first-order fundamental quantities of the S_i. Now, substituting Eq. (5.15) into Eg. (5.14) we get

$$M^{p,q,r}(\mathcal{G}) = \sum_{i=1}^{N} \int_{T_i} \mathbf{f} \cdot \hat{\mathbf{n}}_i \; |\boldsymbol{\alpha}_i||\boldsymbol{\beta}_i| \; \mathrm{d}u\mathrm{d}v \tag{5.16}$$

with

$$\hat{\mathbf{n}}_i = \frac{S_{i,u} \times S_{i,v}}{\sqrt{\mathbf{E}_i \mathbf{G}_i - \mathbf{F}_i^2}} = \frac{\boldsymbol{\alpha}_i \times \boldsymbol{\beta}_i}{|\boldsymbol{\alpha}_i||\boldsymbol{\beta}_i|} \tag{5.17}$$

and

$$\mathbf{f}(x,y,z)|_{T_i} = \mathbf{f}\left(t_i(u,v), y_i(u,v), z_i(u,v)\right), \tag{5.18}$$

with $x_i(u,v)$, $y_i(u,v)$ and $z_i(u,v)$ are the $x-$, $y-$ and $z-$components of $S_i(u,v)$.

5.2.3 Global sensitivity analysis

In GSA, variability of QoI is measured when all parameters vary over the entire design space. This allows users to evaluate the relative contribution of each parameter to QoI's output variation, which is the focus of the present study. Different sensitivity analyses have been proposed in the literature, such as variance-based (or Sobol's method), density-based sensitivity, elementary effects test (or Morris method), etc. Interested readers can

refer to [78] for a detailed overview of these techniques. Among these methods, variance-based probabilistic methods like Sobol's analysis [133] is suitable for complex nonlinear and non-additive models; therefore, it is well received in different design applications and thus used in the current study. This method investigates how much of the overall variance of QoI is achieved due to the variability of a collection of design parameters. This variance is usually measured with *First-order indices* (or *main effects*) or *total-order indices* (or *total effects*). The former quantifies the direct contribution to QoI variance from an individual parameter over the entire design space. The latter approximates the overall contribution of a parameter considering its direct effect and interactions with all the other design parameters.

Sobol's sensitivity analysis

Sobol's analysis is often classified as a variance-based model-independent method, which is based on the variance decomposition and can handle the underlying non-linearity of QoI under consideration. Under the probabilistic interpretation of elements of $\mathbf{t}$, Y is the output of g with mean ($\mathbb{E}(Y)$) and variance ($V(Y)$). Consider $g(t_1, t_2, \ldots, t_n)$ to be square integrable over $\mathcal{X} \subseteq \mathbb{R}^n$ with Lebesque measure $\mathrm{d}\mathbf{t} = \mathrm{d}t_1 \ldots \mathrm{d}t_n$. The Sobol's SA is based on a decomposition of the model into summands of functions of increasing dimensionality referred to as ANOVA (functional ANalysis Of VAriance) or *Hoeffding-Sobol* decomposition [133], that can be written as

$$g(\mathbf{t}) = g_0 + \sum_{s=1}^{n} \sum_{i_1 < \cdots < i_s} \left(t_{i_1}, \ldots t_{i_s}\right), \tag{5.19}$$

where g_0 is the expectation (mean) of Y defined as

$$g_0 = \mathbb{E}(Y) = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} g(t_1, \ldots, t_n) \prod_{k=1}^{n} p_{t_k}(t_k) dt_k, \tag{5.20}$$

and $g_{i_1,\ldots,i_s}(t_{i_1}, \ldots t_{i_s})$ satisfy the unicity condition

$$\int g_{i_1,\ldots,i_s}(t_{i_1},\ldots t_{i_s})p_{t_{i_1,\ldots,i_2}}(t_{i_1},\ldots,t_{i_s})dt_{i_1}\ldots dt_{i_s}=0,\ s=1,2,\ldots,n. \tag{5.21}$$

The interior sum in Eq. (5.19) can be extended over all different groups of indices $i_1, i_2, \ldots, i_s$ such that $1 \leq i_1 < i_2 < \cdots < i_s \leq n$. With this condition, Eq. (5.19) can be expanded as

$$g(t_1,\ldots,t_n) = g_0 + \sum_{i=1}^{n} g_i(t_i) + \sum_{i=1}^{n}\sum_{j=i+1}^{n} g_{ij}(t_i,t_j) + \cdots + g_{1,\ldots,n}(t_1,\ldots,t_n). \tag{5.22}$$

Eq. (5.22) consists of 2^n terms with each term is assumed to be squared integral over $\mathcal{X}$ with zero average. The terms $g_i(t_i)$, $1 \leq i \leq n$, are functions of a single variable and are the so-called first-order indices (or main effect). Each of them represents the variation in Y due to the change in t_i. The functions of more than one variable, $g_{ij}(t_i, t_j)$, $1 \leq i \leq j \leq n$, are called *interactions* and represent the variation in Y not accounted when t_i and t_j are varies individually. With the condition in Eq. (5.21), all the term in Eq. (5.22) are naturally orthogonal and can be expressed as integrals of $g(t)$ as

$$\begin{aligned} g_i(t_i) &= \int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty} g(t_1,\ldots,t_n)\prod_{k=1,k\neq i}^{n} p_{t_i}(t_i)dt_1\ldots dt_n - g_0 \\ &= \mathbb{E}_{\mathbf{t}_{\sim i}}(Y|t_i) - g_0, \end{aligned} \tag{5.23a}$$

$$\begin{aligned} g_{ij}(t_i,t_j) &= \int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty} g(t_1,\ldots,t_n)\prod_{k=1,k\neq ij}^{n} p_{t_i}(t_i)dt_1\ldots dt_n - g_i - g_j - g_0 \\ &= \mathbb{E}_{\mathbf{t}_{\sim ij}}(Y|t_i,t_j) - g_i - g_j - g_0. \end{aligned} \tag{5.23b}$$

In the similar way, Eq.(5.23) continues for the higher-orders. Here, $\mathbb{E}_{\mathbf{t}_{\sim i}}(\cdot)$ is the mean of Y taken over all possible values of $\mathbf{t}$ when t_i is fixed through its full distribution range, whereas $\mathbb{E}_{\mathbf{t}_{\sim ij}}(\cdot)$ is also the mean of Y but evaluated when both t_i and t_j are fixed.

With the hypothesis that all the input parameters are independent of each other, the variance of the output ($V(Y)$) can be also be decomposed into $2^n - 1$ partial variances of increasing orders as [134]

$$V(Y) = \sum_{i}^{n} V_i + \sum_{i} \sum_{j=i+1}^{n} V_{ij} + \cdots + V_{12\ldots n} \tag{5.24}$$

where

$$V_i = V\left(g_i\left(t_i\right)\right) = V_{t_i}\left(\mathbb{E}_{\mathbf{t}_{\sim i}}\left(Y|t_i\right)\right), \tag{5.25a}$$

$$\begin{aligned} V_{ij} &= V\left(g_{ij}\left(t_i, t_j\right)\right) \\ &= V_{t_i,t_j}\left(\mathbb{E}_{\mathbf{t}_{\sim ij}}\left(Y|t_i, t_j\right)\right) - V_{t_i}\left(\mathbb{E}_{\mathbf{t}_{\sim i}}\left(Y|t_i\right)\right) - V_{t_j}\left(\mathbb{E}_{\mathbf{t}_{\sim j}}\left(Y|t_j\right)\right). \end{aligned} \tag{5.25b}$$

Herein, $V_{t_i}(\cdot)$ and $V_{t_j}(\cdot)$ is the variance over all possible values of t_i and t_j, respectively. The contribution of individual design parameter's variance to the total output variance can be evaluated with the above relation. Therefore, by the dividing Eq. (5.23) with the total variance $V(Y)$ of Y one could determine the first and second-order sensitivity index of t_i as

$$SI_i = \frac{V_i}{V(Y)} = \frac{V_{t_i}\left(\mathbb{E}_{\mathbf{t}_{\sim i}}\left(Y|t_i\right)\right)}{V(Y)}, \tag{5.26a}$$

$$SI_{ij} = \frac{V_{ij}}{V(Y)} = \frac{V_{t_i,t_j}\left(\mathbb{E}_{\mathbf{t}_{\sim ij}}\left(Y|t_i, t_j\right)\right) - V_{t_i}\left(\mathbb{E}_{\mathbf{t}_{\sim i}}\left(Y|t_i\right)\right) - V_{t_j}\left(\mathbb{E}_{\mathbf{t}_{\sim j}}\left(Y|t_j\right)\right)}{V(Y)}. \tag{5.26b}$$

Likewise, indices of the $s^{th}-$order can be defined as

$$SI_{i_1,i_2,\ldots,i_s} = \frac{V_{i_1,i_2,\ldots,i_s}}{V(Y)}. \tag{5.26c}$$

SI_i is the main effect index of t_i and can also be referred to as the average reduction of the total variance of Y when t_i is fixed over its full distribution range. Another well known variance-based sensitivity measure is the total effect sensitivity index [134], which can be

derived as

$$SI_{Ti} = \frac{\mathbb{E}_{\mathbf{t}\sim i}\left(V_{t_i}\left(Y|\mathbf{t}_{\sim i}\right)\right)}{V(Y)} = 1 - \frac{V_{\mathbf{t}\sim i}\left(\mathbb{E}_{t_i}\left(Y|\mathbf{t}_{\sim i}\right)\right)}{V(Y)}. \tag{5.27}$$

Here, SI_{Ti} is the total sensitivity index for t_i and $\mathbb{E}_{\mathbf{t}\sim i}\left(V_{t_i}\left(Y|\mathbf{t}_{\sim i}\right)\right)$ is the expected reduction in variance that is obtained if all, but t_i, parameters are fixed. The lower value of SI_{Ti} represents t_i is less significant. Furthermore, the indices in Eq. (5.26) satisfy $\sum_i SI_i + \sum_i \sum_{j>i} SI_{ij} + \cdots + SI_{12\ldots n} = 1$ and sum of the indices in Eq. (5.26a) is greater than or equal to one. In this analysis, if $SI_i = SI_{Ti}$ then there is no interaction effect between t_i and other elements of $\mathbf{t}$ and model is additive, which, based on the assumption of orthogonality of input parameters. If a model is not additive then Sobol's indices can also be used for identifying the effective dimensions [135].

In summary, SI_i as the main effect measures the fractional contribution of a single parameter to the output variance. SI_{ij} are used to measure the fractional contribution of parameter interactions to the output variance. The total effect, SI_{Ti}, is more adequate as its evaluation takes into account the main, second-order, and higher-order effects over the entire range of $\mathcal{X}$ [134]. Therefore, in this work, we focus on evaluating SI_{Ti} of parameters with respect to $\mathcal{MI}^s$. However, as shown above, the analytical evaluation of SI_{Ti} requires solving high-dimensional integrals; therefore, as explained in §3.2.4, sampling methods are used.

5.2.4 Sensitivity analysis of multivariate output

$\mathcal{MI}^s$ of $s^{th}-$order is a vector quantity composed of multiple moments invariant vector terms. For instance, $\mathcal{MI}^2$ is composed of one zeroth-order and six second-order moment invariants, which create a problem with the multivariate output. The typical Sobol decomposition is obtained for each component of the model output, leading to many sensitivity measures for each output variable. These sensitivities can be redundant if the correlation in the model output is essential, leading to difficulties interpreting these results. To deal with

this problem, two different alternatives have been proposed in the literature for multivariate output, referred to as output decomposition [136] and covariance decomposition [122] approaches.

Output decomposition method

The output decomposition method was initially proposed by Campbell et al. [136] and is based on the eigendecomposition of a set of output variables into a lower-dimensional representation. Therefore, it is primarily suitable for problems involving time series output data in which the dimensionality of model output is extensively high. Since $\mathcal{MI}^s$ is the QoI in the present case, we assume that the elements of the $\mathcal{MI}^s$ form a moment space from which a dataset $\mathbb{MII}$ consisting of N' samples is constructed as

$$\mathbb{MII} = \begin{bmatrix} \mathcal{MI}^s_1 \\ \mathcal{MI}^s_2 \\ \vdots \\ \mathcal{MI}^s_{N'} \end{bmatrix} = \begin{bmatrix} \mathbf{MI}^0 & \mathbf{MI}^2_1 & \mathbf{MI}^3_1 & \dots & \mathbf{MI}^s_1 \\ \mathbf{MI}^0_2 & \mathbf{MI}^2_2 & \mathbf{MI}^3_2 & \dots & \mathbf{MI}^s_2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{MI}^0_{N'} & \mathbf{MI}^2_{N'} & \mathbf{MI}^3_{N'} & \dots & \mathbf{MI}^s_{N'} \end{bmatrix}. \tag{5.28}$$

Let $\mathcal{C}$ represents covariance matrix of $\mathbb{MII}$ defined as

$$\mathcal{C} = \frac{1}{N'} \mathbb{MII}_c^T \mathbb{MII}_c, \tag{5.29}$$

where $\mathbb{MII}_c$ is centered matrix obtained by subtracting mean of each column, $\boldsymbol{\mu}$, of $\mathbb{MII}$, i.e., $\mathbb{MII}_c = \mathbb{MII} - \boldsymbol{\mu}$. Now, to identify the orthogonal active directions of moment space, the eigenvectors are computed via their eigenvalue decomposition, which can be written as

$$\mathcal{C} = \mathbf{W} \boldsymbol{\Lambda} \mathbf{W}^T. \tag{5.30}$$

Herein, $\mathbf{W} = \{\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_s\}$ is an $[s \times s]$ matrix whose columns are orthogonal eigenvectors ($\mathbf{w}_k \in \mathbb{R}^{1\times s}$), which spans the new basis to form an eigenspace. Moreover, $\boldsymbol{\Lambda} = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_s)$, with λ_i's being the eigenvalues sorted in descending order $\lambda_1 \geq$

$\lambda_2 \geq \cdots \geq \lambda_s$, which represents the variance resolved along the corresponding eigenvectors. Based on the above decomposition and the variance-based SA, one can use Lamboni et al.'s [137] generalised sensitivity indices for multivariate outputs using the eigenmodes or principal components ($\mathbb{DI}$) obtained with first K eigenvectors which covers at least 95% of the empirical variance, i.e., $\mathbb{DI} = (\mathbb{MI} + \boldsymbol{\mu})\{\mathbf{W}\}_{k=0}^{K}$. The generalised first-order sensitivity index for the i^{th} variable is defined as

$$GSI_i = \sum_{k=1}^{K} \frac{\lambda_k}{V(Y)} SI_{i,k}, \tag{5.31}$$

and the generalised total-order sensitivity index for the i^{th} variable is evaluated as

$$GSI_{T_i} = \sum_{\omega_i} GSI_{\omega_i}, \tag{5.32}$$

where ω_i includes all the components in the ANOVA decomposition with all subscripts including i. $SI_{i,k}$ in Eq. (5.31) are the first-order sensitivity indices of the i^{th} variable, evaluated as in Eq. (5.26a), on the new orthogonal basis $w_{i,k}$. Under this new setting it can be written as

$$SI_{i,k} = \frac{V_{i,k}}{V_k}, \tag{5.33}$$

where $V_{i,k}$ is the partial variance of the k^{th} eigenmode caused by the variation in the i^{th} parameter and V_k is equal to the eigenvalue λ_k. The generalised sensitivity indices in Eq. (5.32) give the significance of the parameters for $\mathcal{MI}^s$ in the same way as the sensitivity indices do in the univariate output case in Eq. (5.27). For more details, interested readers should refer to [136, 137].

Covariance decomposition approach

Gamboa et al. [122] proposed the covariance decomposition approached, which is based on the Hoeffding-Sobol decomposition as in Eq. 5.22. It can be generalised for any arbitrary

number of output variables,

$$\begin{aligned}\mathbf{MI} =&\mathbf{MI}_0^r + \sum_{i=1}^{n} \mathbf{MI}_i^r(t_i) + \sum_{i=1}^{n}\sum_{j=i+1}^{n} \mathbf{MI}_{ij}^r(t_i,t_j) + \cdots + \mathbf{MI}_{1,\ldots,n}^r(t_1,\ldots,t_n),\\ & r = 1,2,\ldots,s,\end{aligned} \tag{5.34}$$

which implies that the covariance matrix of the model output can be partitioned into a sum of covariance matrices as follows:

$$\begin{aligned}\mathcal{C}\left(\mathcal{MI}^s\right) = \mathcal{C}\left(\mathbf{MI}^0,\ldots,\mathbf{MI}^s\right) &= \sum_{i=1}^{n} \mathcal{C}_i\left(\mathbf{MI}^0,\ldots,\mathbf{MI}^s\right)\\ &+ \sum_{i=1}^{n}\sum_{j=i+1}^{n} \mathcal{C}_{i,j}\left(\mathbf{MI}^0,\ldots,\mathbf{MI}^s\right) + \cdots + \mathcal{C}_{1,2,\ldots,n}\left(\mathbf{MI}^0,\ldots,\mathbf{MI}^s\right).\end{aligned} \tag{5.35}$$

The above equation is equivalent to the decomposition of variance in Eq. (5.23) which is used for the scalar output. This implies that the main effect indices can be obtained as $SI_i = V_i/V(M^{0,0,0}) = \mathcal{C}_i/\mathbf{C}(M^{0,0,0})$. Gamboa et al. used this idea to project $\mathcal{C}$ onto a scalar through multiplication by an identity matrix and then taking its trace (Tr) as

$$\begin{aligned}\mathrm{Tr}\left[\mathcal{C}\left(\mathcal{MI}^s\right)\right] = \mathrm{Tr}\left[\mathcal{C}\left(\mathbf{MI}^0,\ldots,\mathbf{MI}^s\right)\right] &= \sum_{i=1}^{n} \mathrm{Tr}\left[\mathcal{C}_i\left(\mathbf{MI}^0,\ldots,\mathbf{MI}^s\right)\right]\\ + \sum_{i=1}^{n}\sum_{j=i+1}^{n} \mathrm{Tr}\left[\mathcal{C}_{i,j}\left(\mathbf{MI}^0,\ldots,\mathbf{MI}^s\right)\right] &+ \cdots + \mathrm{Tr}\left[\mathcal{C}_{1,2,\ldots,n}\left(\mathbf{MI}^0,\ldots,\mathbf{MI}^s\right)\right].\end{aligned} \tag{5.36}$$

On the basis of the above, the multivariate main effect indices of the i^{th} variable can be obtained as

$$GSI_i\left(\mathcal{MI}^s\right) = \frac{\mathrm{Tr}\left[\mathcal{C}_i\right]}{\mathrm{Tr}\left[\mathcal{C}\right]}, \tag{5.37}$$

while the multivariate total effect sensitivity indices are given as

$$GSI_{T_i}\left(\mathcal{MI}^s\right) = \frac{\mathrm{Tr}\left[\mathcal{C}_i\right] + \sum_{i=1}^{n}\sum_{j=i+1}^{n}\mathrm{Tr}\left[\mathcal{C}_{ij}\right] + \cdots + \mathrm{Tr}\left[\mathcal{C}_{1,2,\ldots,n}\right]}{\mathrm{Tr}\left[\mathcal{C}\right]}. \tag{5.38}$$

As the trace, $\mathrm{Tr}\left[\mathcal{C}\left(\mathcal{MI}^s\right)\right]$ is equal to the sum of variances of all elements of $\mathcal{MI}^s$ in Eq. (5.36), GSI_i can be interpreted as the expected percentage reduction in the total variance of the outputs, which is obtained when variable t_i is kept fixed. Garcia-Cabrejo and Valocchi [138] also demonstrated that if the covariance of $\mathcal{MI}^s$ is fully captured by the first K eigenvectors, then GSI_i obtained from this method are equal to GSI_i from *output decomposition approach* in the previous subsection. We will base our SA using the covariance decomposition approach in the present case. Unlike in time series data, the dimensionality of moment space is not extensively high in our case. Furthermore, the covariance decomposition approach reduces the possibility of approximation error during the dimension reduction and numerical inaccuracies resulting from using the output decomposition method during eigendecomposition.

5.2.5 Selection of sensitive parameters

After obtaining the sensitivity indices, a subset of the highly sensitive parameters, whose variation influence significantly the QoI while ignoring those that do not contribute significantly toward design improvement against QoI. Therefore, only a small subset of sensitive parameters are allowed to vary during shape optimisation, and others are kept fixed, thereby accelerating the shape optimisation process. The selection of a subset of sensitive parameters can be made either based on the available computational budget [18,135], using a predefined threshold value (such as ϵ) [72,139] or clustering the parameters into groups of high and low sensitive ones based on their sensitivity indices [37,119]. In the first approach, for instance, if only a limited number of design evaluations are allowed to be performed during optimisation, then the designer will favour selecting a smaller subset of only highly sensitive parameters to achieve maximum possible design improvement within

the available computational budget. A threshold ϵ is defined based on a statistically significant value in the second approach. Any parameter with sensitivity indices greater or equal to ϵ is included in the analysis, and others are kept fixed. The second approach is more favourable and widely practised among these three approaches. It provides the subset of sensitive parameters that are statistically sufficient to redefine the problem with a smaller set of parameters. However, the setting of ϵ is important as the smaller value may result in selecting more parameters, and a larger value may form the subset containing fewer sensitive parameters. For complex analyses, $\epsilon = 0.05$ is widely used [139]. The influence of ϵ on the selection of significantly sensitive parameters will be analysed in §5.4.

5.3 Test cases

To experimentally demonstrate geometric moments' capability to make an informed decision regarding the parametric sensitivity, we use the wave resistance coefficient, C_w, as a physical criterion. C_w is part of the overall resistance affecting the movement of objects on or near the free surface of oceans, lakes and rivers. It reflects the energy spend for creating the free-surface waves following the moving body. Although the overall resistance of the ship is composed of different components, C_w is a vital component and especially prominent for relatively full hull forms travelling at high speeds. It is noteworthy that C_w is highly sensitive to local features of the hull so that a significant reduction can be achieved without affecting the overall cargo capacity. Similar to geometric moments, C_w is affected by the distribution of the hull's shape, and it can be used as a physics-informed shape signature. Minimising this resistance at the preliminary design stage is crucial, but its evaluation can be computationally demanding.

5.3.1 Relation of moments with wave resistance coefficient

Our motivation to investigate the utility of moments in SA for ship design stems from the extensive use of *SAC (Sectional Area Curve)* in Computer-Aided Ship Design and

Hydrodynamic Analysis. SAC is a function $S(x)$ of 2D zeroth-order moments describing the longitudinal variation of the area of ship sections below the waterline. In [140], it is stated: *"A SAC provides an effective and simple description of global geometric properties. At the same time, it is closely related to the resistance and propulsion performance of a ship. From this point of view, the ship hull form distortion approach based on SAC transformation is one of the most effective global design methods for the preliminary design stage."* In analogous spirit, [141] stresses that *"geometric properties of SAC have a decisive effect on the global hydrodynamic properties of ships"*. Historically, the importance of SAC in ship design has been initiated back in the 1950s with the introduction of Lackenby transformation [65] for modifying SAC, which has been further enriched in the context of modern CAD representations and used in ship-design optimisation, see, e.g., in [142, 143].

Furthermore, linear wave-resistance analysis performed by eminent hydrodynamicists, e.g., E.O. Tuck [144,145], J.V. Wehausen [146], has revealed the importance of the longitudinal rate of change of cross-sectional area, i.e., $S'(x)$, which determines the strength of the Kelvin-source distribution used to model the disturbance caused by the body as it moves on the sea's free-surface. It is worth noticing that the flow around a slender ship moving on the free surface with a constant velocity can be represented by using an appropriate source-sink distribution along its centre plane. The strength of these sources is proportional to the longitudinal rate of change of the ship's cross-sectional area [144, 146], and this aspect can be well captured by geometric moments, especially those of higher order. In fact, an early derivation for the evaluation of C_w for slender ships, known as Vosser's integral, reveals explicit dependence on the longitudinal derivative of the cross-sectional area [146], i.e., $S'(x) = \frac{\mathrm{d}}{\mathrm{d}x}S(x)$ where $S(x) = \int_{\Omega(x)} \mathrm{d}y\mathrm{d}z$ is the cross-sectional area, and $\Omega(x)$ denotes the cross-section of a ship hull at the longitudinal position x. Let now $m_p = \int_o^L x^p S'(x)\mathrm{d}x$ be the $p-$th order moment of $S'(x)$ with $x = 0$ and $x = L$ corresponding to the stern and bow tips of the hull, respectively. Assuming that $S(0) = S(L) = 0$ we get:

$$m^p = -p\int_0^L x^{p-1}S(x)\mathrm{d}x = -p\int_0^L \int_{\Omega(x)} x^{p-1}\mathrm{d}x\mathrm{d}y\mathrm{d}z, \tag{5.39}$$

which leads to

$$m^p = -pM^{p-1,0,0}, \tag{5.40}$$

where $M^{p-1,0,0}$ is a component of the hull's geometric moments vector of order $s = p + q + r = p - 1$ (see Eq. (5.2)). Thus, $p-$order 1D moments of $S'(x)$ are directly linked to $(p-1)-$order 3D longitudinal moments of the hull. These physics-informed moments are included in the set of moments used for building the SSV we use for SA.

5.3.2 Parametric modellers

To validate our claim regarding geometric moments, we used two different test cases based on a different type of parameterisation, namely Procedural Deformation (PD) [1, 4] and Global Modification Function (GMF) [12]. PD is used for the parameterisation of the hull shown in Fig. 5.2(a), which shares some closeness to the well known KCS[1] ship hull model, and shall be referred to as the *PD hull* from this point forward in this chapter. GMF is used to parametrise a DTMB 5415[2] naval ship model (see Fig. 5.2(b)), an early and open to the public version of the USS Arleigh Burke destroyer DDG 51, which is another extensively used benchmark ship model for shape optimisation problems.

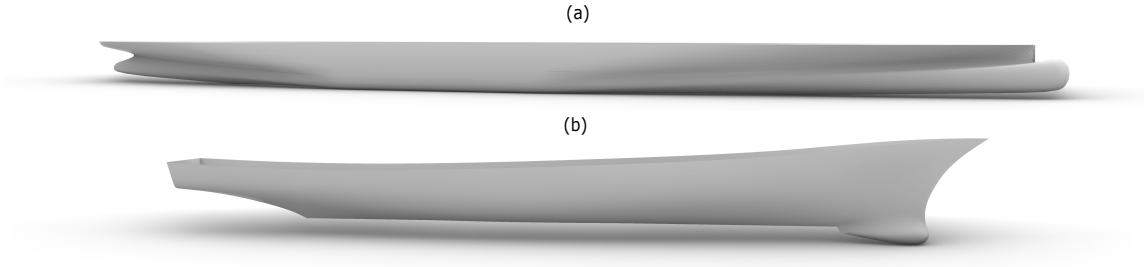


Figure 5.2: Three-dimensional CAD geometries of (a) PD and (b) DTMB 5415 hull models used as test cases for the proposed approach.

PD creates a high-level parameterisation via coupling free-form features with control points of the underlying surface representation through linear procedural relations. This

[1] http://www.simman2008.dk/KCS/container.html
[2] http://www.simman2008.dk/5415/combatant.html

results in a fully feature-driven parameterisation, i.e., each parameter defines and alters a specific feature of the PD hull, such as the length, width or length of the bulbous bow. This parameterisation provides both local and global shape modification. GMF is a grid modification approach performed using a shape modification function based on vector-valued functions defined on a design grid. These functions are defined with the objective that during modification, the underlying structure of the design should be preserved, the design grid used for simulation to evaluate C_w does not have to regenerate, and a prescribed degree of similarity should be maintained. However, unlike PD, the parameterisation obtained from GMF is not feature-driven as varying a parameter may alter a specific feature and features in its neighbourhood. Variations of hull designs obtained from both types of parameterisation are shown in Fig. 5.3.

Procedural deformation (PD)

Let $\mathcal{G}$ be a member of a rich class of objects in an ambient spce $\mathcal{A} \subseteq \mathbb{R}^3$. The PD-based parametric modeller, $\mathcal{P}$, used in this work is based on the technique proposed by Kostas et al. [1], which for any $\mathbf{t} \in \mathcal{X}$ produces a new shape $\mathcal{G}' = \mathcal{P}(\mathbf{t})$. In this case, $\mathcal{P}(\mathbf{t})$ is a vector function $\mathbb{R}^n \to \mathcal{A}$ that defines the underlying geometry of $\mathcal{G}$, which corresponds to $\mathbf{t} = t_i, i = 1, 2, \ldots, 26$. As the PD in this case adopts the NURBS (Non-Uniform Rational B-splines) surface representation, $\mathcal{P}(\mathbf{t}) = N(CG(\mathbf{t}); \Omega)$, where $N : \mathbb{R}^2 \to \mathbb{R}^3$ is a vector-valued function that maps each point of the two dimensional domain, Ω, to a point on the surface bounding $\mathcal{G}$. Here, $CG(\mathbf{t})$ represents the *control cage* of $\mathcal{G}$, which maps $\mathbf{t}$ onto the control points of CG. Parameters in $\mathbf{t}$ are classified in four categories, namely global, local, semi-global and shape transition parameters, providing shape modification of different nature. The parametric definition on the hull geometry is shown in Fig. 5.4.

The global parameters, hull length at waterline, beam and depth, are the most shape influential. Typically, these parameters are predefined during the design process based on the customer requirements; therefore, these are kept fixed in our analysis. The local parameters, such as parameters defining lengths at flat side and bottom at the aft and

Figure 5.3: Three dimensional design variations of the PD hull (on the left) and DTMB 5415 hull (on the right) generated with 26 and 27 design parameters defined using PD- and GMF-based parameterisation, respectively. These design variation can also be visualised in a video at `https://bit.ly/3BiB9wZ`. For PD hull, parametrisation is performed on the submerged part below the waterline, and for both hull their geometric moments are evaluated for the submerged part.

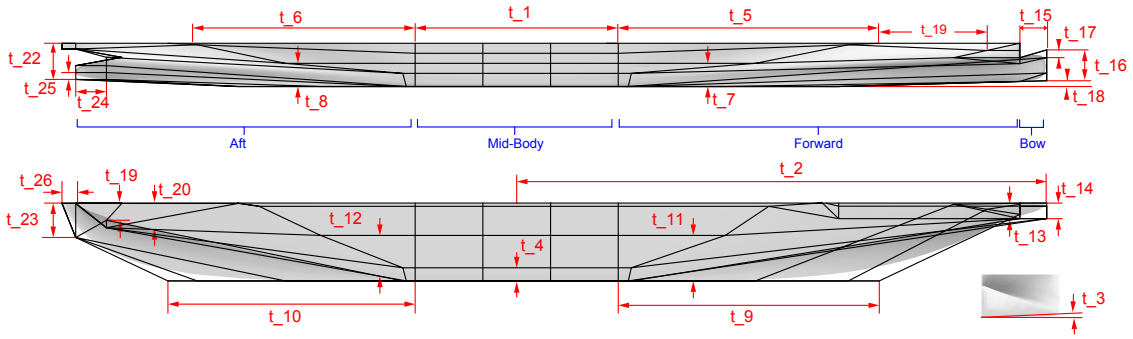


Figure 5.4: Parametrisation of PD hull adopted by the PD-based parametric modeller.

forward (t_5, t_6, t_9, t_{10}), bulb bow (t_{14} - t_{17}), stern (t_{22} - t_{24}) , bilge radius (t_4), shaft (t_{25}, t_{26}) dimensions, etc., affect small areas of the geometry but can cause significant changes on C_w [1]. Semi-global parameters affect relatively large hull areas, such as the length and position of the mid-ship body (t_1 and t_2). Transition parameters are also local, controlling the transition between different sections of the shape, for instance, the transition from mid-ship to bow (t_{19}). In this case, all these parameters are defined as

$$t_i = \hat{t}_i \cdot f_i\left(t_1, t_2, \ldots t_{26}\right), \quad \hat{\mathbf{t}} \in [0,1],\ i = 1, 2, \ldots, 26, \tag{5.41}$$

where $\hat{t}_i$ is the i^{th} non-dimensional version of t_i bounded by [0,1] and f_i are affine functions of these parameters defining the procedural relation and is specified internally by the developer. During shape modification, [1] recommends using $\hat{\mathbf{t}}$ for supporting robustness by avoiding setting parametric values that would result in creating invalid and implausible geometries. Once the values of global parameters and $\hat{\mathbf{t}}$ are given, the control cage, shown in Fig. 5.4, is automatically constructed. For further details on the formulation of this parameterisation, interested readers should refer to [1].

Global modification function (GMF)

Let be given a set of cordinates $\boldsymbol{\zeta} \in \mathcal{G} \subset \mathbb{R}^{\overline{n}}$, with $\overline{n} = 1, 2, 3$, and assume that the design variables set $\mathbf{t}$ defines a continous shape modification vector $\boldsymbol{\delta}(\boldsymbol{\zeta}, \mathbf{t}) \in \mathbb{R}^{\hat{n}}$, with $\hat{n} = 1, 2, 3$,

which for any $\mathbf{t} \in \mathcal{X}$ modifies each $\boldsymbol{\zeta} \in \mathcal{G}$ of the baseline shape to a new shape $\boldsymbol{\zeta}' \in \mathcal{G}'$ as

$$\boldsymbol{\zeta}' = \boldsymbol{\zeta} + \boldsymbol{\delta}(\boldsymbol{\zeta}, \mathbf{t}), \tag{5.42}$$

where $\mathcal{G}'$ is the modified version of the $\mathcal{G}$. In the present work, $\boldsymbol{\delta}(\boldsymbol{\zeta}, \mathbf{t})$ is defined using a recursive combination of $n = 27$ shape modification vectors over a hyper-rectangle embedding the demi hull:

$$\boldsymbol{\psi}_i(\boldsymbol{\zeta}) \ : \ \mathcal{A} = [0, L_{\zeta_1}] \times [0, L_{\zeta_2}] \times [0, L_{\zeta_3}] \in \mathbb{R}^3 \longrightarrow \mathbb{R}^3, \tag{5.43}$$

with $i = 1, \ldots, n$. Specifically,

$$\boldsymbol{\delta}(\boldsymbol{\zeta}, \mathbf{t}) = \boldsymbol{\delta}_n, \tag{5.44}$$

where

$$\boldsymbol{\delta}_i(\boldsymbol{\zeta}, \mathbf{t}) = t_i \boldsymbol{\psi}_i(\boldsymbol{\zeta}), \quad \text{with} \quad \begin{cases} \boldsymbol{\zeta} = \boldsymbol{\zeta} + \boldsymbol{\delta}_{i-1} \\ \boldsymbol{\delta}_1 = 0 \end{cases} \tag{5.45}$$

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

$$\boldsymbol{\psi}_i(\boldsymbol{\zeta}) := \prod_{j=1}^{3} \sin\left(\frac{a_{ij}\pi\zeta_j}{L_{\zeta_j}} + r_{ij}\right) \mathbf{e}_{q(i)}. \tag{5.46}$$

In Eq. (5.46), $\{a_{ij}, j = 1, 2, 3\} \in \mathbb{R}$ define the degree of the function along j-th axis, $\{r_{ij}, j = 1, 2, 3\} \in \mathbb{R}$ are the corresponding spatial phases and $\{L_{\zeta_j}, j = 1, 2, 3\} \in \mathbb{R}$ 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 [12].

The objective of using two different types of parameterisation on two different hulls is to see if geometric moment invariants capture the sensitivity of parameters under various design settings.

5.3.3 Hydrodynamic solver and setup

Two inviscid solvers are for two ship hulls, both symmetric with respect to the xz-plane. For the one hull referred to as the PD hull model, the PD parameterisation is used, while for the other hull, referred to in the literature as the DTMB hull, we employ the GMF parameterisation.

PD hull model: Hydrodynamic calculations to estimate C_w for the PD hull are performed using the Isogeometric Analysis-based Boundary Element Method (IGA-BEM) developed by Belibassakis et al. [15]. This solver applies the Isogeometric Analysis (IGA) [147] for solving the boundary integral equation (BIE) associated with the linearised Neumann–Kelvin formulation for the calculation of C_w of ships. The IGA concept is based on exploiting the same NURBS basis used to represent the exact geometry of the hull for approximating the singularity distribution of the associated BIE, or, in general, the dependent physical quantities. In BIE, the dependent/unknown variable is the density of Neumann Kelvin sources distributed over the hull's wetted surface, which is accurately represented with parametric NURBS surfaces or a collection of smoothly joined NURBS patches; referred to as multi-patch NURBS surface. In our case, the PD hull is composed of a signal cubic NURBS surface with 108 control points, whose iso-mesh is shown in Fig. 5.5 and simulation is performed on the unit scaled right demi-hull.

DTMB hull model: Hydrodynamic simulations of this hull model are conducted using the code WARP (Wave Resistance Program), developed at CNR-INM. C_w computations are based on linear potential flow theory using Dawson (double-model) linearisation. The frictional resistance is estimated using a flat-plate approximation based on the local Reynolds number. Details of equations, numerical implementations, and validation of the numerical solver are given in [120]. As with the DTMB hull model, simulations are performed for the demi-hull. Figure 5.5 (b) shows the computational grid used for the simulation. The computational domain for the free-surface is defined within $1L_{\text{pp}}$ upstream, $3L_{\text{pp}}$ downstream, and $1.5L_{\text{pp}}$ sideways. A total of 75×20 grid nodes are used for the free

surface, whereas 90×25 nodes are used for the hull discretisation. The main particulars, test conditions and C_w values for both type hulls are given in Table 5.1.

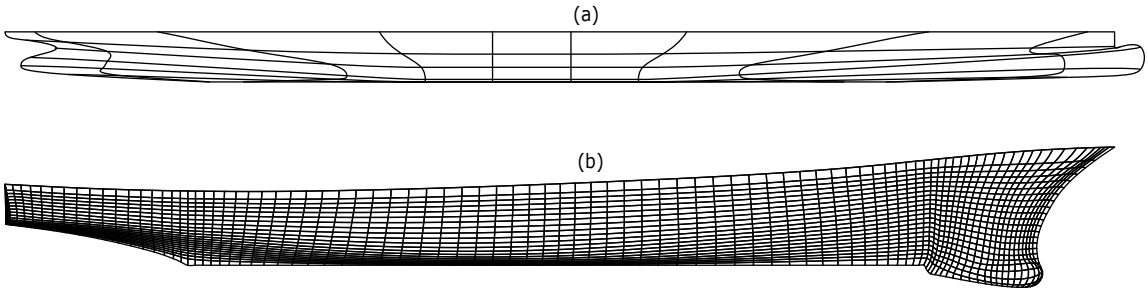


Figure 5.5: Computational grid of (a) PD hull and (b) DTMB 51415 hull used during the simulation for approximation of C_w.

Table 5.1: Main particular, test conditions and C_w values of PD hull and DTMB 5415 hull.

Quantity	Symbol	Unit	Value	
			DTMB Hull	PD Hull
Volume of displacement	∇	m^3	8419.31	5112.56
Wetted Surface area	S	m^2	2974.23	2076.56
Length at waterline	Lwl	m	142.73	100.00
Max Beam at waterline	Bwl	m	19.06	16.25
Draft	T	m	6.15	4.51
Water density	ρ	kg/m^3	998.50	
Kinematic viscosity	ν	m^2/s	1.09E-6	
Gravity acceleration	g	m/s^2	9.80	
Froude number	Fr	–	0.25	
Wave resistance coefficient	C_w	–	1.0531E-03	1.0678E-04

5.4 Results and discussion

This section demonstrates the effectiveness of geometric moment invariants for evaluating parametric sensitivities using various experiments on the previously described test cases. We first provide the results of geometric moment invariants for PD and DTMB hull models and then discuss the Sensitivity Analysis (SA) results with respect to Shape-Signature

Vector (SSV) and C_w for both hull types, along with the correlation between the results. Finally, for both test cases, we present the optimisation results performed in the sensitivity spaces evaluated with SSV and C_w.

5.4.1 Moment Evaluation

As mentioned before, for any shape satisfying the conditions mentioned in Section 5.2.2, there exist geometric moments of arbitrary order. In this work, we mainly focus on performing SA with respect to fourth-order geometric moments invariant to the translation and scaling. This choice results from two facts:

1. Higher-order moments can be sensitive to noise acquisition [118]. The risk of numerical inaccuracies, specifically due to the use of finite-precision arithmetics, also increases as we move towards evaluating high-order moments [116]. Therefore, it could be challenging to include moments of order greater than 10 as computational complexity increases with the order.

2. Literature review in various application areas, e.g., kinetic equations [148] and shape retrieval [132], reveals that it is unlikely to use moments of order higher than 4.

Thus, SSV of order $s = 4$ ($\mathcal{MI}^4$) is used to evaluate the sensitivity of parameters of both test cases. In $\mathcal{MI}^4$, there are 1, 6, 10 and 15 components of $0^{th}-$ $2^{nd}-$, $3^{rd}-$ and $4^{th}-$order geometric moments, respectively. The values of these invariants for two hulls are shown in Table 5.2. As explained earlier in Section 5.2, all the moment invariants of first-order are zero, while the zeroth-order moment is equal to one due to its scale invariance. Therefore, the effect of sensitivity of parameters in the case of zeroth-order is measured with $M^{0,0,0}$ instead of $MI^{0,0,0}$, as $M^{0,0,0}$ defines the volume of the shape.

5.4.2 Sensitivity Analysis of PD hull model

As described earlier, the PD hull is parameterised with 26 procedural parameters, so a 26-dimensional design space is created while keeping the baseline hull at the centroid of

Table 5.2: Geometric moment invariants up to $4^{th}-$order evaluated for the PD and DTMB hull models.

Designs	$M^{0,0,0}$	$MI^{2,0,0}$	$MI^{0,2,0}$	$MI^{0,0,2}$	$MI^{1,1,0}$
DTMB Hull	8.4193E+03	2.3151	4.1970E-02	6.9840E-03	0
PD Hull	5.1126E+03	1.7426	5.7126E-02	5.4962E-03	0
	$MI^{1,0,1}$	$MI^{0,1,1}$	$MI^{0,0,3}$	$MI^{0,1,2}$	$MI^{0,2,1}$
DTMB Hull	-2.3789E-02	0	-3.3039E-04	0	1.0767E-03
PD Hull	-3.8124E-03	0	-4.7635E-05	0	3.7467E-04
	$MI^{0,3,0}$	$MI^{1,0,2}$	$MI^{1,1,1}$	$MI^{1,2,0}$	$MI^{2,0,1}$
DTMB Hull	0	2.7862E-03	0	-9.0788E-03	2.4529E-03
PD Hull	0	-6.2258E-05	0	-4.8513E-03	1.7167E-02
	$MI^{2,1,0}$	$MI^{3,0,0}$	$MI^{0,0,4}$	$MI^{0,1,3}$	$MI^{0,2,2}$
DTMB Hull	0	4.4042E-01	1.3333E-04	0	2.2588E-04
PD Hull	0	1.5272E-01	5.5021E-05	0	3.1210E-04
	$MI^{0,3,1}$	$MI^{0,4,0}$	$MI^{1,0,3}$	$MI^{1,1,2}$	$MI^{1,2,1}$
DTMB Hull	0	3.9970E-03	-8.8414E-04	0	-5.5388E-04
PD Hull	0	6.7697E-03	-2.8814E-05	0	-1.8780E-04
	$MI^{1,3,0}$	$MI^{2,0,2}$	$MI^{2,1,1}$	$MI^{2,2,0}$	$MI^{3,0,1}$
DTMB Hull	0	2.2982E-02	0	6.0453E-02	-2.2388E-01
PD Hull	0	9.2384E-03	0	6.3636E-02	-2.3872E-02
	$MI^{3,1,0}$	$MI^{4,0,0}$			
DTMB Hull	0	12.3709			
PD Hull	0	6.5083			

the design space. To commence the SA, the design space is sampled with $N' = 9000$ samples using a progressive sampling technique [18, 38]. This sampling technique is based on the space-filling criterion, searching the design space in the class of Latin-Hypercube to ensure a uniformly distributed and diverse set of samples. $\mathcal{MI}^4$ and C_w of designs are evaluated on the basis of the setting described in Section 5.2.2 and 5.3.3, respectively. Afterwards, two different datasets are created, the first containing the design parameter values as independent variables and C_w as dependent variables. The second dataset is composed of $\mathcal{MI}^4$ as dependent variables. Afterwards, Sobol's global SA for univariate output is performed to measure the sensitivity of parameters towards C_w, and multivariate output Sobol's analysis with covariance decomposition approach is utilised to estimate the parameters' sensitivity to $\mathcal{MI}^4$. In the remainder of this section, we first discuss the results on the sensitivity of parameters with respect to C_w (shown in Fig. 5.6) and then we present the results of parameters' sensitivity measured with respect to the zeroth- to the fifth-order (shown in Fig. 5.7 and 5.8) SSV to observe how the sensitivity of parameters varies with the increment in the order. Finally, we compare the sensitivity indices of parameters evaluated with $\mathcal{MI}^4$ and C_w.

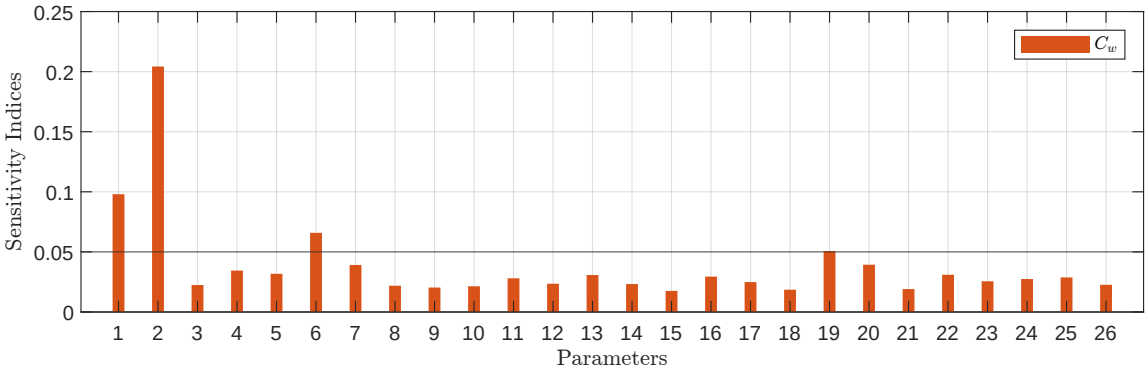


Figure 5.6: Sensitivity indices of PD hull's 26 design parameters obtained using Eq. (5.27) with respect to C_w.

To select the set of sensitive parameters, a threshold is set equal to $\epsilon = 0.05$ [139]. From Fig. 5.6, it can be seen that in case of C_w, 4 out of 26 parameters, t_1, t_2, t_6 and t_{19},

have a sensitivity index greater than ϵ and thus regarded as the C_w sensitive parameters. Among these parameters, t_1 and t_2 have a substantially higher sensitivity index, defining the mid-body length and position. These parameters are of semi-global nature and have the highest impact on the hull's shape after the three global parameters, and thus, they significantly affect the variation of C_w. The next two sensitive parameters, t_6 and t_{19}, have a sensitivity index close to ϵ. Here, t_6 is a local parameter, which modifies the flat-of-side length at the aft of the hull and t_{19} is the transition parameter defining the interaction of the bow and bulb of the hull. These results align with the literature [18] as parameters associated around the bulbous bow segment of the hull are known to have a significant influence on C_w.

The parametric sensitivity indices obtained with SSV are shown in Fig. 5.7. We will start the discussion with the results related to $\mathcal{MI}^0$, which only consists of the zeroth-order moment, $\mathbf{M}^0$, i.e., the volume of the submerged part of the hull. In this case, there are three parameters, t_1, t_4 and t_6, sensitive to $\mathcal{MI}^0$. The reader should recall that, among these parameters, t_1 and t_6 are sensitive to C_w, which means that $\mathcal{MI}^0$ is able to capture 50% of the parametric sensitivity to C_w. Obviously, the parameter t_1 is more sensitive to volume as it modifies the length of the mid-body of the hull. However, as the volume does not get affected by changing the position of the mid-ship, t_2, which is a third sensitive parameter with regards to C_w, has a negligible effect on $\mathcal{MI}^0$. Interestingly, in the case of $\mathcal{MI}^2$, similar with respect to C_w, there are four parameters, t_1, t_2, t_5 and t_6, with indices higher than 0.05 and out of these four sensitive parameters, three parameters, t_1, t_2 and t_6, are also sensitive to C_w. Note that the sensitivity index of t_2 is now close to that obtained with C_w, which means that $\mathcal{MI}^2$ can capture the sensitivity of the shape caused by varying mid-body position (i.e., the parameter t_2). More importantly, the parameters, t_1, t_2 and t_6 are the top three most sensitive parameters both with respect to C_w and $\mathcal{MI}^2$. The parameter, t_5, which is only sensitive to $\mathcal{MI}^2$, is local by definition and modifies the flat-of-side at the forward part of the hull.

From the results depicted in Fig. 5.7, it can be seen that via $\mathcal{MI}^2$ we are able to capture

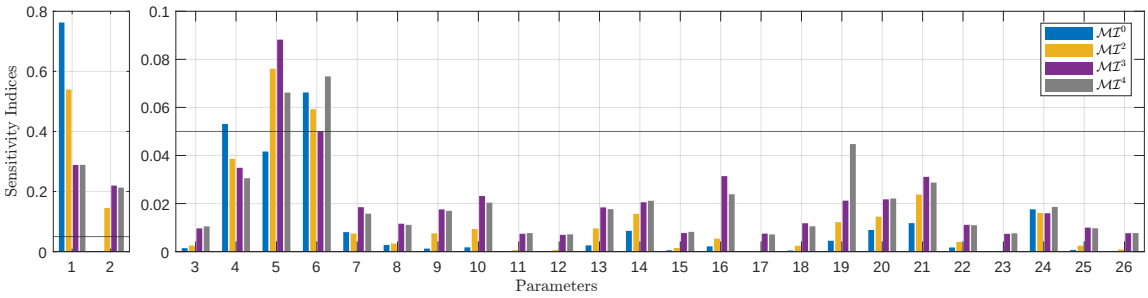


Figure 5.7: Sensitivity indices of PD hull's 26 design parameters obtained using Eq. (5.38) with respect to $\mathcal{MI}^0$, $\mathcal{MI}^2$, $\mathcal{MI}^3$ and $\mathcal{MI}^4$.

the sensitivity of not only the semi-global parameters (t_1, t_2) but also the sensitivity of the local parameters (t_5, t_6), which means the in comparison with $\mathcal{MI}^0$, $\mathcal{MI}^2$ is more capable of reflecting the sensitivity of parameters with respect to C_w. For the transition parameter, t_{19}, its sensitivity index has increased, but it is still far from being categorised as sensitive. Therefore, geometric moment invariants of higher order may be required.

In the case of $\mathcal{MI}^3$ and $\mathcal{MI}^4$, the sensitive parameters remain the same as in the case of $\mathcal{MI}^2$. However, the sensitivity indices of almost the entire set of parameters differ from what was obtained with $\mathcal{MI}^0$ and $\mathcal{MI}^2$. As we moved from $\mathcal{MI}^0$ to $\mathcal{MI}^4$, the domination of sensitivity indices of highly sensitive parameters (t_1, t_2) reduces and sensitivity indices of other parameters increases. Primarily, this is prominent for t_1, whose sensitivity indices decreases significantly from 0.7625 ($\mathcal{MI}^0$) to 0.2889 ($\mathcal{MI}^4$). However, from $\mathcal{MI}^3$ to $\mathcal{MI}^4$ sensitivity indices remain similar. Another essential point to note here is that, when sensitivity analyses are performed with $\mathcal{MI}^4$, the sensitivity index of transition parameter t_{19}, which is sensitive with respect to C_w and is the only parameter that could not be categorised as sensitive to geometric moments, increases monotonically as 0.0045 ($\mathcal{MI}^0$), 0.0122 ($\mathcal{MI}^2$), 0.0212 ($\mathcal{MI}^3$) and 0.0447 ($\mathcal{MI}^4$). It can be seen that at $\mathcal{MI}^4$, the sensitivity index of t_{19} becomes very close to one obtained with C_w. This indicates that as we move towards higher-order geometric moments, we capture more detailed geometric information of the hull, including its local features. To further validate this, we increased the order of SSV from 4 to 5 and analysed the behaviour of parametric

sensitivities with the inclusion of 5th order moments in SSV (i.e., $\mathcal{MI}^5$). It can be seen from Fig. 5.8 that parameters sensitive with respect to $\mathcal{MI}^4$ are also sensitive to $\mathcal{MI}^5$. However, the sensitivity indices obtained with $\mathcal{MI}^5$ tends to be similar to ones obtained with C_w. Summery of these sensitivity results is provided in Table 5.3.

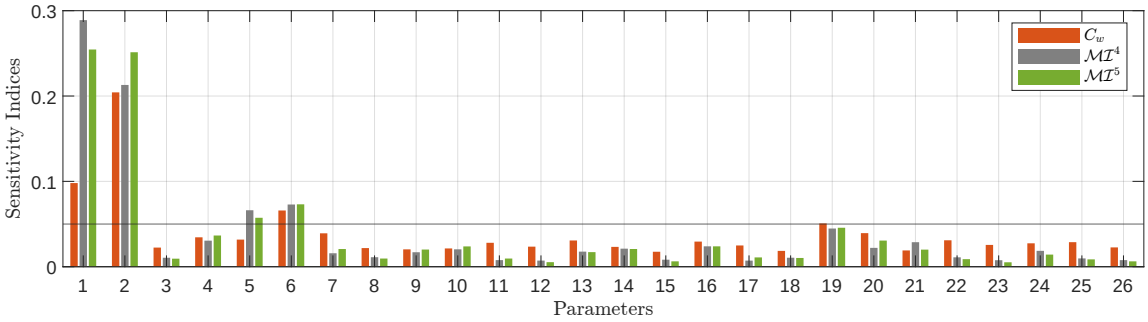


Figure 5.8: Sensitivity indices of PD hull's 26 design parameters obtained with respect to 4th and 5th order SSV (i.e., $\mathcal{MI}^4$ and $\mathcal{MI}^5$) and wave resistant coefficient (C_w).

Table 5.3: Sensitive parameters of PD hull with respect to C_w and $\mathcal{MI}^s$ with $s = 0/2/3/4/5$.

QoI	Sensitive parameters	NMSE	Similarity
C_w	t_1, t_2, t_6, t_{19}	-	-
$\mathcal{MI}^0$	t_1, t_4, t_6	0.7399	58%
$\mathcal{MI}^2$	t_1, t_2, t_5, t_6	0.4822	75%
$\mathcal{MI}^3$	t_1, t_2, t_5, t_6	0.2221	75%
$\mathcal{MI}^4$	t_1, t_2, t_5, t_6	0.2146	75%
$\mathcal{MI}^5$	t_1, t_2, t_5, t_6	0.1856	75%

Note the correlation results discussed above are evaluated at $\epsilon = 0.05$. As stated in §5.2.5, when the value of ϵ changes the then parameters sensitive to both C_w and $\mathcal{MI}^s$ change. For instance, in Fig. 5.6, when we set $\epsilon = 0.1$, we will have only one parameter in the subset of significantly sensitive parameters and setting $\epsilon = 0.075/0.05/0.04/0.03$ will result in 2/4/4/10 sensitive parameters, respectively, in the final subset. Similarly, in Fig. 5.7, when ϵ is equal to 0.04 and 0.05 then parameters sensitive to $\mathcal{MI}^0$ are $[t_1, t_4, t_5, t_6]$ and $[t_1, t_4, t_6]$, respectively. In both cases, $[t_1, t_4]$ are also sensitive to C_w. So,

respectively of the specific value adopted for ϵ, one can make a good preliminary estimation of sensitive parameters at a significantly reduced computational cost. A commonly used threshold value is $\epsilon = 0.05$ [72, 139], specifically for selecting a subset of significantly sensitive parameters to construct a design space of reduced dimension. This is the value used in this work.

Moreover, as the parametric sensitivities are evaluated based on SDD [18], we also perform an experiment where the number of samples is varied against the sensitivity indices of parameters. Fig. 5.9 show the sensitivity indices of the top 5 sensitive parameters (t_1, t_2, t_4, t_5 and t_6) evaluated with respect to $\mathcal{MI}^4$ versus the number of samples. It can be seen that sensitivity indices vary as the sample size increases, especially for parameter t_1, which is the topmost sensitive parameter. However, sensitivity indices reach a plateau after a sample size equal to 1000, sufficient to reach true parametric sensitivities.

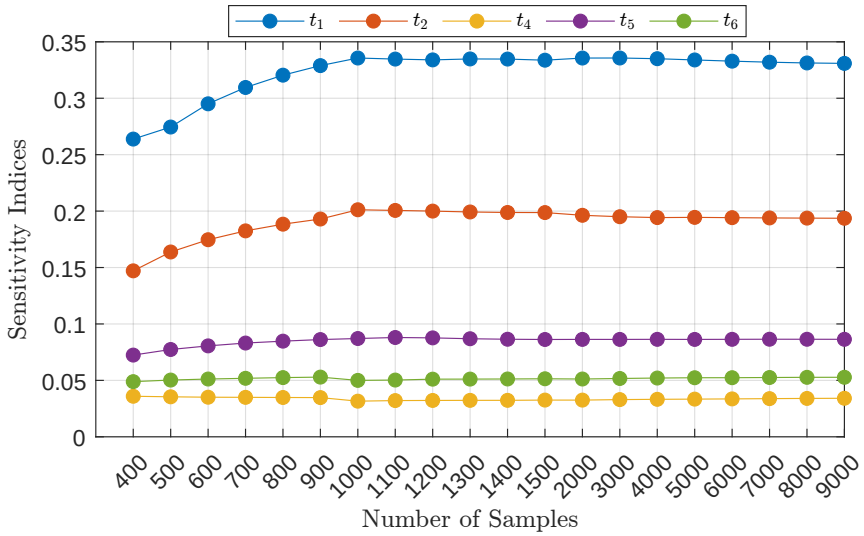


Figure 5.9: Plot showing sensitivity indices evaluated with respect to $\mathcal{MI}^4$ versus the number of samples used to perform the geometric moment-dependent sensitivity analysis.

Metric to measure correlation

To further analyse the effect of sensitivity of parameters with respect to geometric moments and to compare it versus the sensitivity indices to that obtained for C_w, we introduce two metrics as below:

$$\text{NRMSE} = \frac{\sqrt{\sum_{i=1}^{n} \frac{\left(GSI_{T_i} - SI_{T_i}\right)^2}{n}}}{\max\left(\mathbf{SI}_T\right) - \min\left(\mathbf{SI}_T\right)} \tag{5.47}$$

$$\text{similarity} = \frac{\overline{\mathbf{SI}}_T \cdot \overline{\mathbf{GSI}}_T}{||\overline{\mathbf{SI}}_T|| \; ||\overline{\mathbf{GSI}}_T||} = \frac{\sum_i^n \overline{SI}_{T_i} \; \overline{GSI}_{T_i}}{\sqrt{\sum_i^n \overline{SI}_{T_i}} \; \sqrt{\sum_i^n \overline{GSI}_{T_i}}} \tag{5.48a}$$

where

$$\overline{SI}_{T_i} = \begin{cases} 1 & \text{if} \quad SI_{T_i} \geq \epsilon \\ 0 & \text{otherwise} \end{cases}, \quad \overline{GSI}_{T_i} = \begin{cases} 1 & \text{if} \quad GSI_{T_i} \geq \epsilon \\ 0 & \text{otherwise} \end{cases}. \tag{5.48b}$$

The first metric (Eq. (5.47)) is the normalised root mean squared error (NRMSE) which works directly on the sensitivity indices and measures the deviation between the two sets of sensitivity indices evaluated with C_w ($\mathbf{SI}_T$) and $\mathcal{MI}^s$ ($\mathbf{GSI}_T$). The second metric (Eq. (5.48)) is based on the cosine similarity, bounded between [0,1], which is used to measure the similarity between the parameters sensitive to $\mathcal{MI}^s$ and C_w. The perfect scenario will be that the parameters sensitive to C_w are also sensitive to $\mathcal{MI}^s$ or vice versa. Therefore, we measure the similarity on the two binary sensitivity vectors evaluated with C_w ($\overline{\mathbf{SI}}_T$) and $\mathcal{MI}^s$ ($\overline{\mathbf{GSI}}_T$) obtained with Eq. (5.48b). A sensitive parameter, i.e., a parameter with a sensitivity index greater than or equal to 0.05, gets a score of one, and an insensitive parameter gets zero. This is preliminary because we are not interested in the sensitivity indices; instead, we intend to categorise a parameter as sensitive or insensitive according to the sensitivity indices using the set threshold, ϵ.

Figures 5.10(a, b) and Table 5.3 depict the NRMSE and similarity values obtained using Eq. (5.47) and (5.48) for $\mathcal{MI}^0$, $\mathcal{MI}^2$, $\mathcal{MI}^3$, $\mathcal{MI}^4$ and $\mathcal{MI}^5$. From these figures, it can

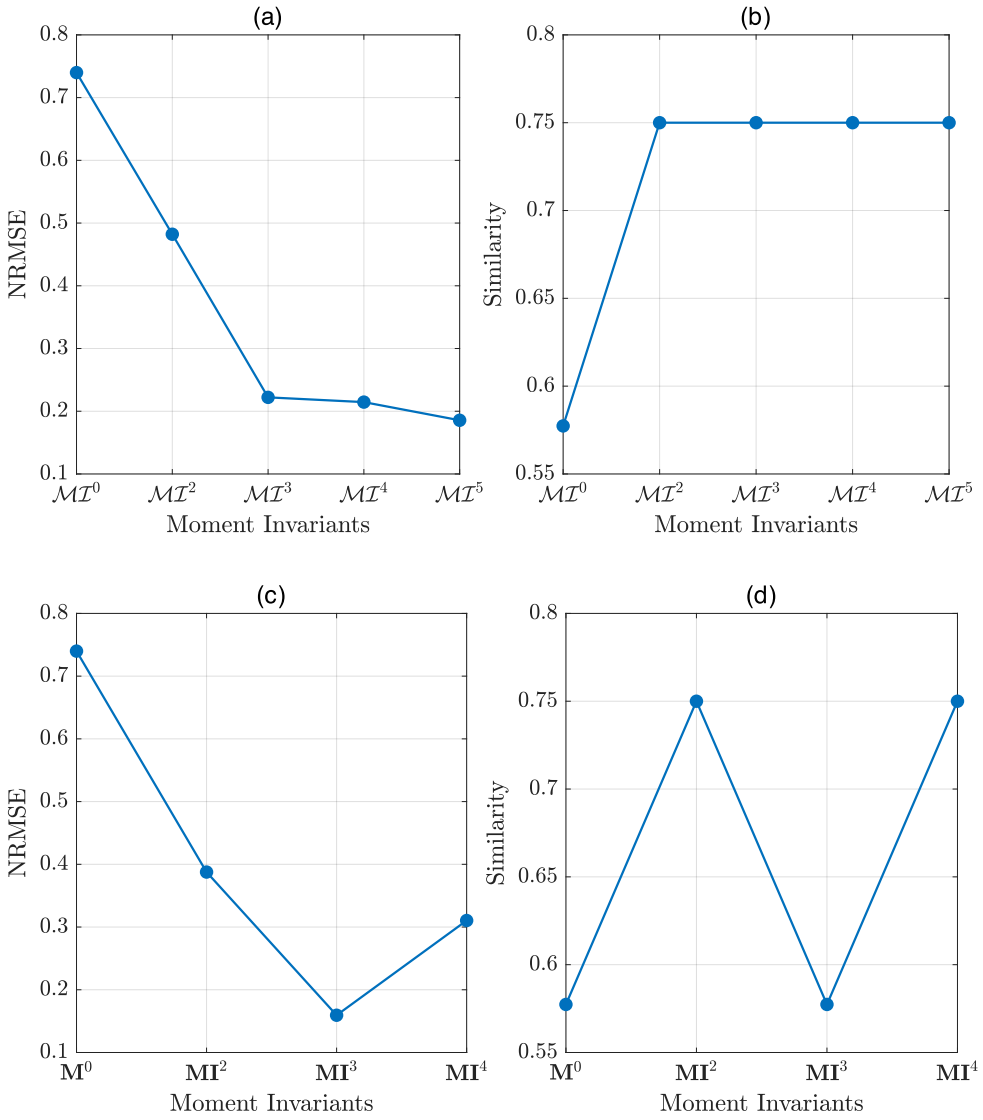


Figure 5.10: Plot showing (a, c) NRMSE and (b, d) similarity values obtained using Eq. (5.48) and (5.47) for $\mathcal{MI}^0$ to $\mathcal{MI}^5$ and $\mathbf{MI}^0$ to $\mathbf{MI}^4$ obtained for PD hull.

be seen that NRMSE achieves its maximum for $\mathcal{MI}^0$ and has a steep descent up to $\mathcal{MI}^3$. There is no significant difference between NRMSE of $\mathcal{MI}^3$ and $\mathcal{MI}^4$; however, it reduces slightly from 0.2146 to 0.1856 when sensitivity analyses are performed with $\mathcal{MI}^4$ and $\mathcal{MI}^5$. This is an apparent behaviour occurring as we increase the order of SSV by adding high-order geometric moments. The sensitivity indices of the parameters become closer to the sensitivity indices obtained with C_w. As mentioned earlier, sensitive parameters identified by $\mathcal{MI}^2$, $\mathcal{MI}^3$, $\mathcal{MI}^4$ and $\mathcal{MI}^5$ are the same, i.e., in all three cases, there is a 0.75 (or 75%) similarity between the sensitive parameters of SSVs and C_w. This shows that adding higher-order moments to SSV can better capture the parametric sensitivities; however, the improvement is marginal.

So far, we have measured the sensitivity of the parameters using $\mathcal{MI}^s$, which contains all the geometric moments from up to $s^{th}-$order. The question arises what would be the result if we use geometric moment invariants of a particular order to perform SA, i.e., if we measure sensitivity to $\mathbf{MI}^s$ which contains the moments of $s^{th}-$order only. The results of this experiment are shown in terms of NRMSE and similarity in Figures 5.10(c) and (d), respectively. At $\mathbf{M}^0$, NRMSE is the highest, and similarity is the lowest. These values are equal to the case when $\mathcal{MI}^0$ is used as $\mathcal{MI}^0 = \mathbf{M}^0$ due to the reason mentioned earlier in Section 5.2.2. When $\mathbf{MI}^2$ is used to measure the sensitivity of the parameters, interestingly, NRMSE is lower than $\mathcal{MI}^2$, and both have the same similarity to the sensitive parameters of C_w. Similarly, NRMSE obtained with $\mathbf{MI}^3$ is lower than the $\mathcal{MI}^3$; however, the similarity with sensitive parameters of C_w is only 57%, which is the same when $\mathcal{MI}^0$ is used. The NRMSE of sensitivity indices obtained with $\mathbf{MI}^4$ and C_w is 0.3103, which is higher than the ones obtained with $\mathcal{MI}^3$ and $\mathcal{MI}^4$; nevertheless, the similarity is the same as $\mathcal{MI}^4$. These results show fluctuations in the sensitivity indices when only $\mathbf{MI}^s$ are used. This is because for geometric moments to be used as a shape descriptor, the SSV should be composed of all the geometric moments up to a specific order [116].

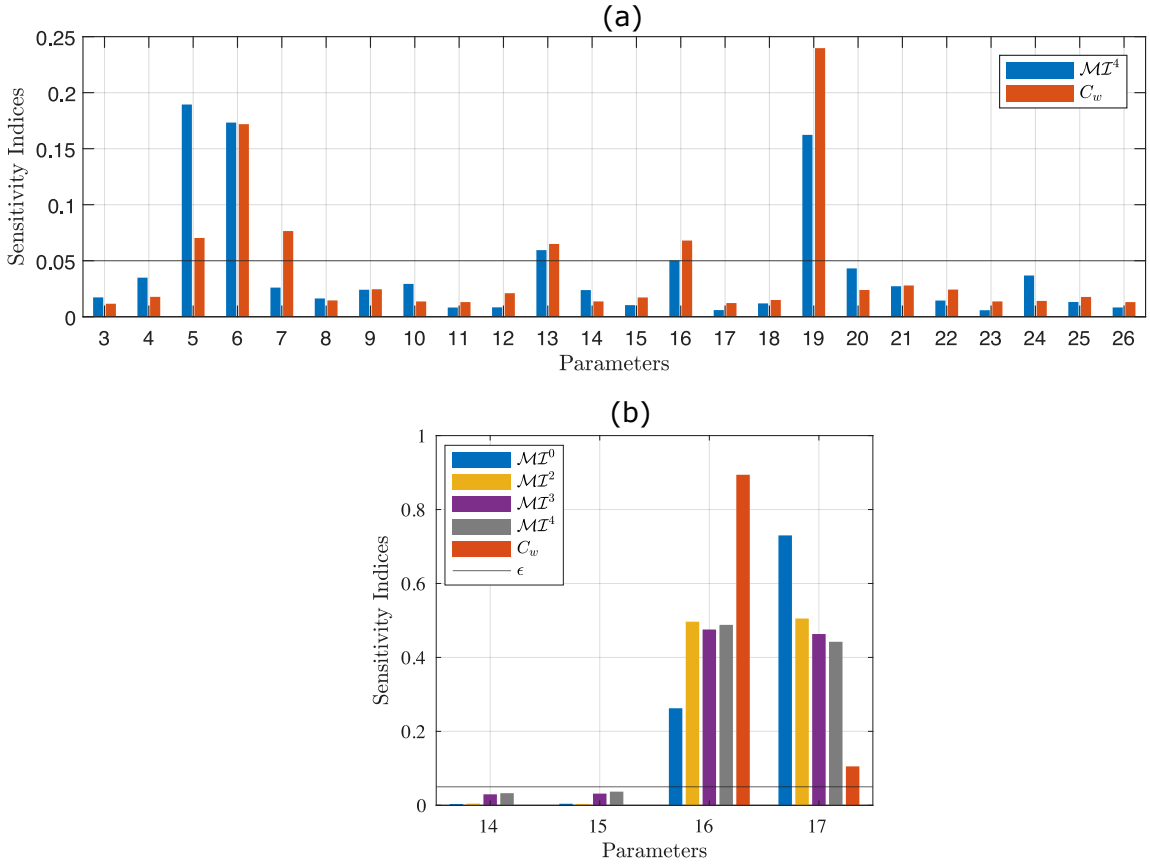


Figure 5.11: Sensitivity indices obtained with sensitivity analysis preformed (a) with only local and transition parameters (t_3 to t_{26}) and (b) with only bow parameters (t_{14} to t_{17}) of PD hull using Eq. (5.27) and (5.38) with respect to C_w and $\mathcal{MI}^s$, respectively.

Sensitivity analysis while excluding most dominating sensitive parameters

In this section we have so far observe that parameters t_1 and t_2 are the most sensitive parameters to C_w, $\mathcal{MI}^2$, $\mathcal{MI}^3$ and $\mathcal{MI}^4$, which, as indicated in Section 5.3.2, are semi-global in nature when it come to their high impact on shape modification. Therefore, it will be interesting to see if we exclude (i.e., keep them fixed) these parameters and perform SA on the remaining 24 parameters with respect to $\mathcal{MI}^4$ and C_w. The results of this experimentation are shown in Fig. 5.11(a).

There is a couple of noteworthy remarks regarding these results. First of all, there are

six and five parameters, [t_5, t_6, t_7, t_{13}, t_{16}, t_{19}] and [t_5, t_6, t_{13}, t_{16}, t_{19}] sensitive to C_w and $\mathcal{MI}^4$, respectively, with $\epsilon \geq 0.05$. In the case of C_w, parameter, t_{19}, is the most sensitive one, followed by t_6. Note that out of these parameters, t_6 and t_{19}, are also sensitive when SA is performed with all 26 parameters; see in Fig. 5.7 and Table 5.4. It is noteworthy that in this case, t_{19} is a third sensitive parameter to $\mathcal{MI}^4$ and unlike the results in Fig. 5.7, it has significant sensitivity with the index value close to that of t_5 and t_6. Furthermore, apart from parameter t_7, all the remaining parameters sensitive to $\mathcal{MI}^4$ are also sensitive to C_w. Therefore, there is a high degree of similarity between sensitive parameters obtained with both quantities with similarity values equal to 0.9129 (91.29%) and NRMSE equal to 0.1296. This shows that by identifying more sensitive parameters with SSV, one can also fix the most dominating sensitive parameters and re-perform the sensitivity study as the computational cost of evaluating the moments is significantly less than evaluating C_w.

Table 5.4: Local and transition sensitive parameters of PD hull with respect to C_w and $\mathcal{MI}^4$.

QoI	Sensitive parameters	NMSE	Similarity
C_w	t_5, t_6, t_7, t_{13}, t_{16}, t_{19}	-	-
$\mathcal{MI}^4$	t_5, t_6, t_{13}, t_{16}, t_{19}	0.1296	91%

5.4.3 Sensitivity analysis for shape with simple geometry

As stated earlier in this section, selecting the order of SSV is influenced by the nature of problems' physics and geometry. It is reasonable to expect that for a simple geometry with fewer complex features, lower-order SSV would be sufficient for SA. To analyse this effect, we perform SA only for the parameters $[t_{14}, t_{15}, t_{16}, t_{17}]$ associated with the width, length, height and tip height of PD hull's bulbous bow. Compared to the overall hull form, the geometry of the bulbous bow is simple (mainly of elliptical type) but critical from the physical point of view. Sensitivity analyses are performed to measure the sensitivity of these parameters with respect to C_w and SSV, $\mathcal{MI}^s$ with $s = 0/2/3/4$ while fixing

other parameters. The results of this experimentation are shown in Fig. 5.11(b). It can be seen that parameters $[t_{16}, t_{17}]$ sensitive to C_w with sensitivity indices greater than $\epsilon = 0.05$, are also sensitive to SSV of all orders. However, for $\mathcal{MI}^0$, the indices of t_{16} and t_{17} deviate largely from what obtained with respect to C_w. As order increases, the index of t_{16} increases whereas index of t_{17} decreases; getting closer to indices obtained with C_w. There is no significant difference between sensitivity indices obtained with respect to $\mathcal{MI}^2$, $\mathcal{MI}^3$ and $\mathcal{MI}^4$. Interestingly, in the case of $\mathcal{MI}^2$, $\mathcal{MI}^3$ and $\mathcal{MI}^4$, the sensitivity ranking of parameters is the same as obtained via C_w; thus, it gives similarity equal to 1.0000 (100%). The results of this experiment reveal that in the case of simple geometry, geometric moments of lower order are enough to capture parametric sensitivities associated with the local features of the hull.

5.4.4 Sensitivity analysis of DTMB hull model

As in the case of the PD hull, SA for the parameters of the DTMB hull commences with a 27-dimensional design space, which was sampled to create a dataset consisting of $N' = 9000$ samples. First, SA is performed to measure the sensitivity of parameters with respect to C_w, which is evaluated using the potential flow solver as described in Section 5.3.3. Afterwards, the covariance decomposition approach, along with Sobol's SA, is used to evaluate generalised total sensitivity indices for the parameters with respect to $\mathcal{MI}^0$, $\mathcal{MI}^2$, $\mathcal{MI}^3$ and $\mathcal{MI}^4$. As the objective here is to preliminary evaluate the parametric sensitivity using geometric moments, we first analyse results in terms of NRMSE and similarity, which are shown in Fig. 5.12(a) and (b), respectively. These results exhibit similar behaviour as the results in the case of the PD hull shown in Fig. 5.10(a) and (b). NMSE obtained with sensitivity indices of C_w and $\mathcal{MI}^0$ is the highest and decreases steadily up to $\mathcal{MI}^4$, with similar values when SA is performed with $\mathcal{MI}^2$ and $\mathcal{MI}^3$. Likewise, with the increase in the order of SSV, the similarity between the sensitive parameters obtained using C_w and geometric moments increases gradually from 0.1690 (16.90%) (evaluated with $\mathcal{MI}^0$) to 0.7715 (77.15%) (evaluated with $\mathcal{MI}^4$). One should recall that in the previous test case,

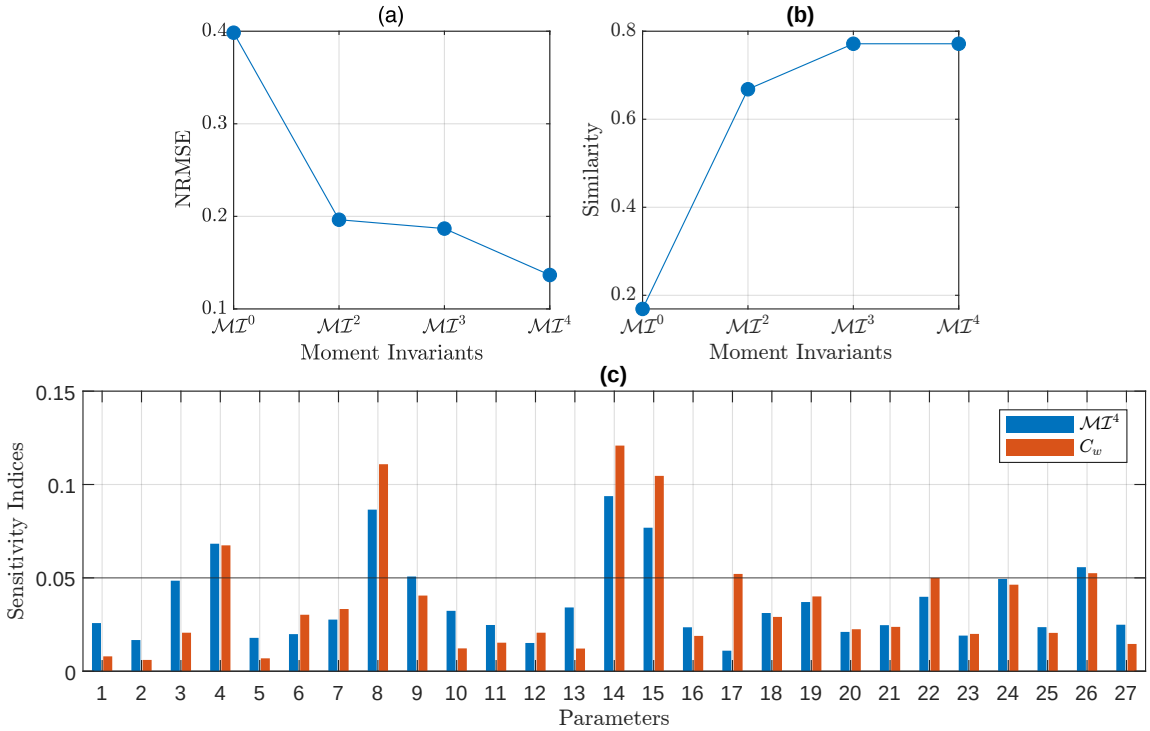


Figure 5.12: Plot showing (a) NRMSE and (b) sensitivity values obtained using Eq. (5.48) and (5.47) for $\mathcal{MI}^0$ to $\mathcal{MI}^4$ obtained for DTMB hull. (c) Sensitivity indices of DTMB hull's 27 design parameters obtained using Eq. (5.27) and (5.38) with respect to C_w and $\mathcal{MI}^4$, respectively.

the similarity value of 0.75 was consistent for $\mathcal{MI}^2$, $\mathcal{MI}^3$ and $\mathcal{MI}^4$, i.e., 75% of parameters sensitive to C_w were identified. However, in this test case, $\mathcal{MI}^2$ could only identify 66.81% of the parameters sensitive to C_w. Nevertheless, the similarity value achieved with $\mathcal{MI}^4$ for this test case is slightly higher than what was obtained for the PD Hull. This shows that also in this test case, $\mathcal{MI}^4$ achieves the highest similarity and lowest NRMSE to sensitivity indices of C_w, whereas $\mathcal{MI}^0$ shows the least similarity and highest NMSE. In conclusion, even for a test case like DTMB, whose parameterisation is not feature-driven, geometric moments can still significantly capture the sensitivity of parameters associated with physics. Summery of above discussed sensitivity results is provided in Table 5.5.

Fig. 5.12(c) show the sensitivity indices of the 27 design parameters obtained with

Table 5.5: Sensitive parameters of DTMB hull with respect to C_w and $\mathcal{MI}^s$ with $s = 0/2/3/4$.

QoI	Sensitive parameters	NMSE	Similarity
C_w	t_4, t_8, t_{14}, t_{15}, t_{17}, t_{22}, t_{26}	-	-
$\mathcal{MI}^0$	t_9, t_{22}, t_{23}, t_{25}, t_{26}	0.3984	17%
$\mathcal{MI}^2$	t_3, t_4, t_8, t_9 t_{14}, t_{15}, t_{24}, t_{26}	0.1964	69%
$\mathcal{MI}^3$	t_4, t_8, t_9 t_{14}, t_{15}, t_{22}, t_{26}	0.1868	77%
$\mathcal{MI}^4$	t_4, t_8, t_9 t_{14}, t_{15}, t_{22}, t_{26}	0.1366	77%

respect to C_w and $\mathcal{MI}^4$. It can be seen that for C_w, 7 out of the 24 parameters, [t_4, t_8, t_{14}, t_{15}, t_{17}, t_{22}, t_{26}], have a sensitivity index greater than ϵ and thus can be regarded as the most sensitive parameters with respect to C_w. Among these parameters, t_{14}, t_8 and t_{15} have substantially high sensitivity index while t_{17}, t_{22} and t_{26} have a sensitivity index close to $\epsilon = 0.05$. In case of $\mathcal{MI}^4$, there are 6 parameters, [t_4, t_8, t_9 t_{14}, t_{15}, t_{22}, t_{26}], with sensitivity index higher than 0.05. It is interesting to note that, except parameter t_9, parameters sensitive to $\mathcal{MI}^4$ are also sensitive to C_w. More importantly, the parameters, t_4, t_{14}, t_8 and t_{15}, are the top 4 sensitive parameters with respect to both C_w and $\mathcal{MI}^4$.

5.4.5 Composite geometric moment invariants

For complex geometries containing many features, SSV may require to include high-order geometric moments to capture detailed information about the shape, mainly associated with local intrinsic features. However, as mentioned earlier, higher-order moments are sensitive to noise. Therefore, instead of evaluating higher-order geometric moments for capturing detailed features, one may decompose the geometry into smaller segments whose geometries are simple enough to be represented easily with lower-order geometric moments. This will create a Composite-Shape-Signature Vector containing the geometric moment invariants up to $s^{th}-$order for all shape segments and then use it to perform SA. In this connection, we shall henceforth refer to composite $\mathcal{MI}^s$ versus the global $\mathcal{MI}^s$ used in the previous section.

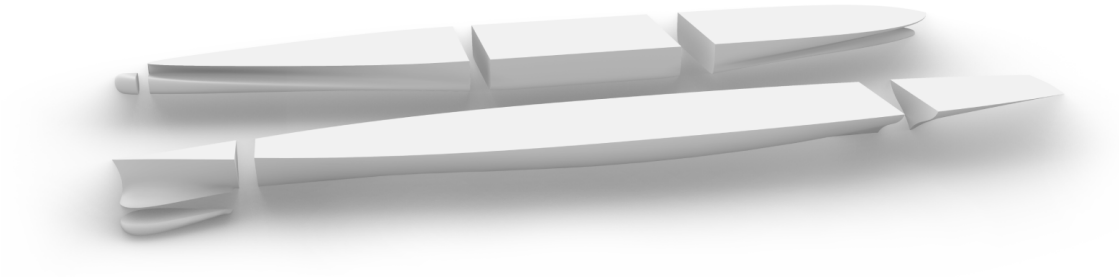


Figure 5.13: Shape segmentation of PD and DTMB hulls used for sensitivity analysis performed with composite $\mathcal{MI}^s$.

The segmentation of ship hulls used as test cases in the present work is shown in Fig. 5.13. Both hulls are divided into four parts: bow (bulbous bow for the PD hull and sonor dome for the DTMB hull), forward, mid-body, and aft segments. This segmentation is relatively easy for ship hulls and widely adopted in the literature. However, for other shapes, one can perform the segmentation based on the visible features or use an automatic segmentation method similar to one presented in [149]. After the segmentation, $\mathcal{MI}^4$ is evaluated, and the multivariate SA is performed, whose sensitivity results, along with C_w, are shown in Fig. 5.14 and 5.15 for PD hull and DTMB hull, respectively. The set of sensitive parameters of both hulls with respect to composite-$\mathcal{MI}^4$ are provided in Table 5.6. We shall first comment on the sensitivity results of the PD hull obtained with composite $\mathcal{MI}^4$ and compare them with the sensitivity results of global $\mathcal{MI}^4$ evaluated for the entire shape, given in Fig. 5.7. In this case, sensitivity results for C_w are the same as previously presented; however, the sensitivity of parameters to geometric moments changes due to the usage of composite $\mathcal{MI}^4$. From these results it can be seen that there are five parameters, $[t_1, t_2, t_5, t_6, t_{19}]$, sensitive to composite $\mathcal{MI}^4$ where, apart from t_{19}, the four remaining parameters are also sensitive with respect to global $\mathcal{MI}^4$. One should recall that in Fig. 5.7, t_{19} is the only parameter sensitive with respect to C_w but insensitive to global $\mathcal{MI}^4$. However, it is noteworthy that in the case of composite $\mathcal{MI}^4$, this parameter is sensitive with a substantial increment in its sensitivity index value. This results in the decline

in NRMSE from 0.2145 (obtained with global $\mathcal{MI}^4$) to 0.1661 (obtained with composite $\mathcal{MI}^4$) and increment in similarity from 0.75 (75%) to 0.8944 (89.44%). Moreover, in the case of composite $\mathcal{MI}^4$, the first two sensitive parameters have the same ranking as the ranking obtained with C_w. This again shows the usability of geometric moments to perform reliable global SA.

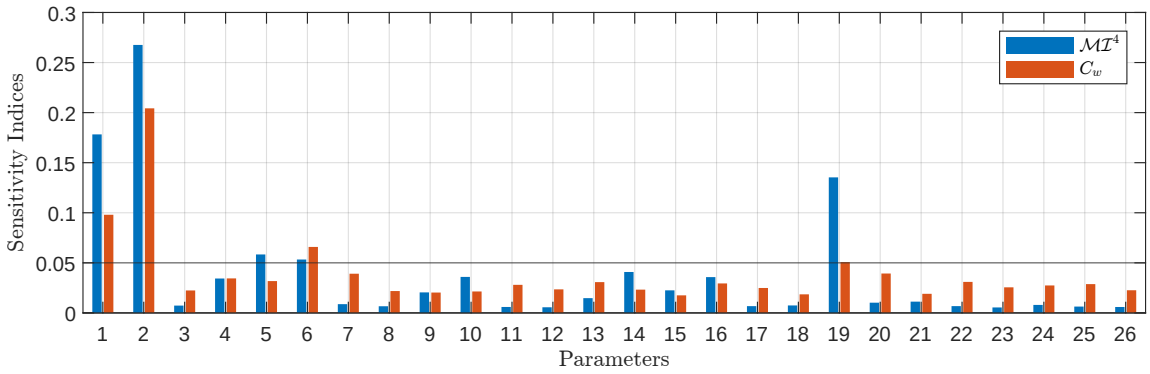


Figure 5.14: Sensitivity indices of PD hull's 26 design parameters obtained using Eq. (5.27) and (5.38) with respect to C_w and composite $\mathcal{MI}^4$, respectively.

Similar to the PD hull, interesting and improved results (shown in Fig. 5.15) are obtained when SA are performed for DTMB hull with composite $\mathcal{MI}^4$. In this case, instead of 7 we have 6 sensitive parameters, $[t_4, t_8, t_{14}, t_{15}, t_{22}, t_{26}]$. All the parameters sensitive to composite $\mathcal{MI}^4$ are also sensitive with respect to C_w. Furthermore, now the similarity between two sets of sensitive parameters increases from 0.7715 (77.15%) to 0.9258 (92.58%) and only one parameter, t_{17}, could not be categorised sensitive with respect to the composite $\mathcal{MI}^4$. The parameters t_9 and t_{24} are sensitive with respect to global $\mathcal{MI}^4$ and insensitive with respect to composite $\mathcal{MI}^4$ with significant reduction in their sensitivity indices. Furthermore, the parameter t_{22} is sensitive with respect to composite $\mathcal{MI}^4$ and C_w, but is insensitive with respect to global $\mathcal{MI}^4$ (see results in Fig. 5.12(c)). Therefore, NRMSE reduces from 0.1366 to 0.1301.

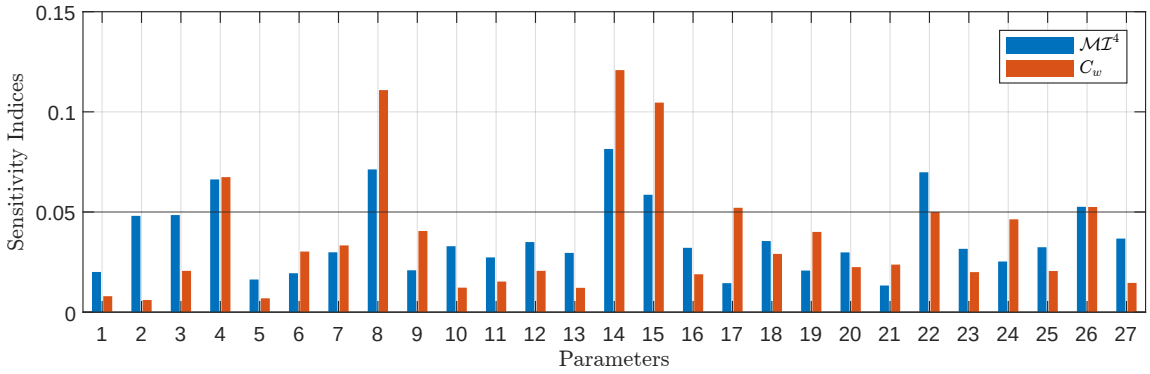


Figure 5.15: Sensitivity indices of DTMB hull's 27 design parameters obtained using Eq. (5.27) and (5.38) with respect to C_w and composite $\mathcal{MI}^4$, respectively.

Table 5.6: Sensitive parameters of PD and DTMB hulls with respect to composite-$\mathcal{MI}^4$.

QoI	Sensitive parameters	NMSE	Similarity
	PD Hull		
Composite-$\mathcal{MI}^4$	t_1, t_2, t_5, t_6, t_{19}	0.1661	89%
	DTMB Hull		
Composite-$\mathcal{MI}^4$	t_4, t_8, t_{14}, t_{15}, t_{22}, t_{26}	0.1301	93%

5.4.6 Selection of SSV's order to commence SA

From the above experimentation, it can be concluded that selecting the right order of SSV can be based on the complexity of the shape's geometry. For a geometry with fewer complex features, it would be sufficient to work with lower-order SSV, e.g., of order 2 or 3. On the contrary, complex geometries with many features may require SSV to include high-order geometric moments (e.g., of order ≥ 4) to capture detailed information about the shape, mainly associated with local intrinsic features. However, the higher-order moments can be sensitive to noise acquisition [118]. Therefore, instead of evaluating higher-order geometric moments for capturing detailed features, one could decompose the geometry into smaller segments whose geometries are simple enough to be represented easily with lower-order SSV. As a rule of thumb, *we recommend using 4th-order global-SSV for a simple geometry and 4th-order composite-SSV for complex geometry with many features.* This is also supported by the fact that various application areas, e.g., kinetic equations [148] and shape retrieval [132], use geometric moments up to the order of 4 and as shown in §5.4.2, with SSV of order higher than 4 only a marginal improvement can be excepted. The above discussion on the selection of the right order of SSV in relation to geometry complexity also aligns with the result of experiments performed in §5.4.2 and 5.4.3. For instance, in the case of PD hull, only composite-SSV could categorise parameter, t_{19}, related to local feature like bulbous bow as sensitive; see Fig. 5.7, 5.8 and 5.14 for comparison. In contrast, for a bulbous bow, whose geometry type is simple compared to the overall hull form, the global-SSV of 2nd-order was enough to capture the parametric sensitives; see Fig. 5.11(b).

5.4.7 Summary of sensitivity results

Before proceeding to the last part of this section, which will exploit our SA approach for shape optimisation, we offer the reader a summary of the sensitivity results obtained:

1. For both the PD and DTMB hulls, the NRMSE between sensitivity indices evaluated with respect to C_w and $\mathcal{MI}^s$ reduces with the increment in s.

2. Similarity between the parameter sensitivities with respect to C_w and $\mathcal{MI}^s$ increases with respect to s up to $s = 3$ but no significant improvement is observed when $s = 4$.

3. In comparison to global $\mathcal{MI}^s$, the composite $\mathcal{MI}^s$ captures better parametric sensitivities.

4. The results in Fig. 5.7 and 5.11 show that out of the 26 parameters that control the parametric modeller of PD, 7/8 parameters are significantly sensitive with respect to $\mathcal{MI}^4/C_w$, respectively (see Table 5.7).

5. A similar behaviour is observed from Fig. 5.12 and 5.15 for the DTMB hull: 7/6 parameters out of the 27 parameters are sensitive with respect to $\mathcal{MI}^4/C_w$, respectively (see Table 5.7).

6. The similarity between the subsets of significantly sensitive parameters with respect to C_w and $\mathcal{MI}^4$ is 79.06% for the PD hull and 92.58% for the DTMB hull.

7. In the case of PD (DTMB) hull, the SA with respect to $\mathcal{MI}^4$ helps to achieve a 73.08% (74.07%) reduction in the dimension of the design space.

Table 5.7: Summary of the sensitivity analysis results obtained from the previously discussed experimentation.

	Sensitive Parameters	Similarity
	with respect to C_w	
PD Hull	$t_1,\ t_2,\ t_5,\ t_6,\ t_7,\ t_{13},\ t_{16},\ t_{19}$	-
DTMB Hull	$t_4,\ t_8,\ t_{14},\ t_{15},\ t_{22},\ t_{26}$	-
	with respect to $\mathcal{MI}^4$	
PD Hull	$t_1,\ t_2,\ t_5,\ t_6,\ t_{13},\ t_{16},\ t_{19}$	79.06%
DTMB Hull	$t_4,\ t_8,\ t_{14},\ t_{15},\ t_{17},\ t_{22},\ t_{26}$	92.58%

5.4.8 Shape optimisation

One of the key objectives of extracting parametric sensitivities is to achieve rapid design improvements already at the preliminary stage of shape optimisation. In this connection, once the subset $\mathbf{t}_{\mathcal{MI}}$ of parameters sensitive to $\mathcal{MI}^4$ is selected from the original parametric set $\mathbf{t}$, we use them to construct a design space $\mathcal{X}_{\mathcal{MI}}$ of lower dimension to expedite the shape optimisation performed against physical QoI, i.e., C_w. For the PD and DTMB hulls, their sensitive parameters $\mathbf{t}_{\mathcal{MI}} = \{t_1, t_2, t_5, t_6, t_{13}, t_{16}, t_{19}\}$ and $\mathbf{t}_{\mathcal{MI}} = \{t_4, t_8, t_{14}, t_{15}, t_{17}, t_{22}, t_{26}\}$ create two 7-dimensional design spaces ($\mathcal{X}_{\mathcal{MI}}$). As stated earlier that typically parametric sensitivities are learnt directly with physical QoI, which can be extensively computational demanding due to the need of performing physical analysis for both SA and shape optimisation. To experimentally prove the potential of moments for learning parametric sensitivities to aid optimisation, we construct a reduced-dimension design space $\mathcal{X}_{c_w}$ with the subset $\mathbf{t}_{c_w}$ of parameters sensitive to C_w. The shape optimisation is performed in $\mathcal{X}_{\mathcal{MI}}$ and $\mathcal{X}_{c_w}$ with an objective to minimise their C_w.

The parametric modellers [1,12] used in this work ensure the generation of valid geometries, i.e., the possibility of generating disjoint and self-intersecting surfaces is negligible. However, a valid geometry may be unrealistic or impractical; therefore, our optimisation in Eq. (5.1) can be reformulated based on the set of design constraints as:

Optimisation:

$$\begin{aligned} &\text{Find } \mathbf{t}^*_{\mathcal{MI}} \in \mathbb{R}^m \quad \text{such that} \\ &\qquad C_w(\mathbf{t}^*_{\mathcal{MI}}) = \min_{\mathbf{t}_{\mathcal{MI}} \in \mathcal{X}_{\mathcal{MI}}} C_w(\mathbf{t}^*_{\mathcal{MI}}) \\ &\qquad \text{subject to} \quad 0.95V_0 \leq V(\mathbf{t}_{\mathcal{MI}}) \leq 1.05V_0, \\ &\qquad\qquad\qquad\quad 0.95Bwl_0 \leq Bwl(\mathbf{t}_{\mathcal{MI}}) \leq 1.05Bwl_0, \\ &\qquad\qquad\qquad\quad Lwl(\mathbf{t}_{\mathcal{MI}}) = Lwl_0, \\ &\qquad\qquad\qquad\quad T(\mathbf{t}_{\mathcal{MI}}) = T_0. \end{aligned} \tag{5.49}$$

Here, $\mathbf{t}_{\mathcal{MI}}$ is a subset of sensitive parameters obtained with respect to moment and $\mathcal{X}_{\mathcal{MI}}$ is the corresponding design space, whose dimension is less than the original space ($\mathcal{X}$). V, B_{WL}, L_{WL}, T correspond to volume, length and beam at the waterline, and draft, respectively. The sub-index $(\cdot_0)$ indicates the quantity values for the baseline hull design. These constraints focus on exploring an optimal design whose key features reside in the vicinity of the baseline design. Therefore, the resulting optimal design is considered practical as the baseline design.

The optimisation is performed using the Jaya Algorithm (JA) [109], with the objective to minimise C_w. JA is a simple yet effective stochastic meta-heuristic optimisation technique whose performance has been proven in various engineering applications. JA may provide different results in each run; therefore, five different optimisation runs are performed in the present work. In each run, a total of 150 iterations are conducted. Table 5.8 shows the C_w values obtained at the 150th iteration of shape optimisation performed in $\mathcal{X}_{c_w}$ and $\mathcal{X}_{\mathcal{MI}}$) for PD and DTMB hulls, along with the average C_w in all five runs. From this table, it can be seen that in the case of the PD hull, the best design is obtained at the fourth and fifth run from $\mathcal{X}_{\mathcal{MI}}$ and $\mathcal{X}_{c_w}$ with C_w equal to 1.0205E-07 and 1.0302E-07, respectively. Fig. 5.16 (a) shows C_w values in all 150 iterations for these two runs. It is noteworthy that the optimal design obtained from $\mathcal{X}_{\mathcal{MI}}$ has better performed (i.e., its C_w value is less) compared to the design obtained from $\mathcal{X}_{c_w}$. However, on average, the C_w obtained from $\mathcal{X}_{\mathcal{MI}}$ with fives run at their 150th iteration is slightly higher, with C_w equal to 1.0335E-07, compared to the design obtained from $\mathcal{X}_{c_w}$, which is equal to 1.0316E-07. The optimal designs obtained for the PD hull from this shape optimisation experiment are shown in Figures 5.17. Figures 5.17(a) and (b) compare the baseline and optimal designs obtained from $\mathcal{X}_{\mathcal{MI}}$ and $\mathcal{X}_{c_w}$ in terms of their cross-sections (bodyplan) while Fig. 5.17(c) show this comparison between the two optimal designs.

Figures 5.17(d-f) show the intensity of deviation between the features of these designs as a heat map plotted using the one-sided Hausdorff Distance (H_d) between two objects and features with maximum deviation ($H_d = 1$) are highlighted in red. Similar features are

highlighted in dark blue when $H_d = 0$. From these results, it can be seen that the design optimised from $\mathcal{X}_{c_w}$ has maximum deviation at the forward part of the hull, and the design optimised from $\mathcal{X}_{\mathcal{MI}}$ show maximum deviation from the baseline design at the aft segment. These slight geometric dissimilarities between the two optimised hulls (see Fig. 5.17 (c) and (f)) are due to the existence of one extra parameter (t_7) in $\mathbf{t}_{c_w}$, which, as shown in Fig. 5.4, modifies the rise of the forward part of hull's flat-of-side. Nevertheless, in terms of performance, there is no significant difference between two hulls; i.e., the C_w of optimised hulls in Fig. 5.17(a) and (b) obtained with $\mathbf{t}_{c_w}$ and $\mathbf{t}_{\mathcal{MI}}$ is 1.2976E-07 and 1.0328E-07, respectively. The C_w value of the baseline PD hull (shown in Fig. 5.1(a)) is 1.07E-04, which is higher than the optimal designs obtained from both sensitive design spaces. This shows that one can achieve significant improvement in the design using sensitive parameters only.

Fig. 5.16 (b) shows the C_w values over 150 iterations during the fourth and fifth optimisation run for the DTMB hull performed in $\mathcal{X}_{\mathcal{MI}}$ and $\mathcal{X}_{c_w}$ created for DTMB hull. From the results in Fig. 5.16 (b) and Table 5.8 it can be seen that, like in the case of PD hull, the optimal design obtained in $\mathcal{X}_{\mathcal{MI}}$ has also lower C_w value compared to the design obtained from $\mathcal{X}_{c_w}$. These values are notably less than that of the baseline design of DTBM hull in Fig. 5.1(b), which is equal to 1.05E-03. Moreover, for this hull, the average C_w value of five optimisation runs, shown in Table 5.8, is 5.0917E-04 when optimisation is performed in $\mathcal{X}_{\mathcal{MI}}$, which, unlike the case of PD hull, is slightly higher then average C_w (equal to 5.0447E-04) obtained from $\mathcal{X}_{c_w}$. The percentage difference between the two values is 0.92%, which is negligible. The comparison between the optimal and baseline design of the DTMB hull is shown in Figure 5.18 in terms of the bodyplan and the Hausdorff distance heat map. The map in Fig. 5.18 (d) shows that in the case of $\mathcal{X}_{\mathcal{MI}}$ the maximum deviation of the optimal design from the baseline occurs close to the waterline at the mid-body segment of the hull. In contrast, the design optimised from $\mathcal{X}_{c_w}$ deviates notably from the baseline design at the forward segment close to the entrance of the hull; Fig. 5.18 (e). On the other hand, comparing the two optimised in Figure 5.18(c) and (f) one can observe a similar behaviour as the PD hull in term of geometric variation. For the

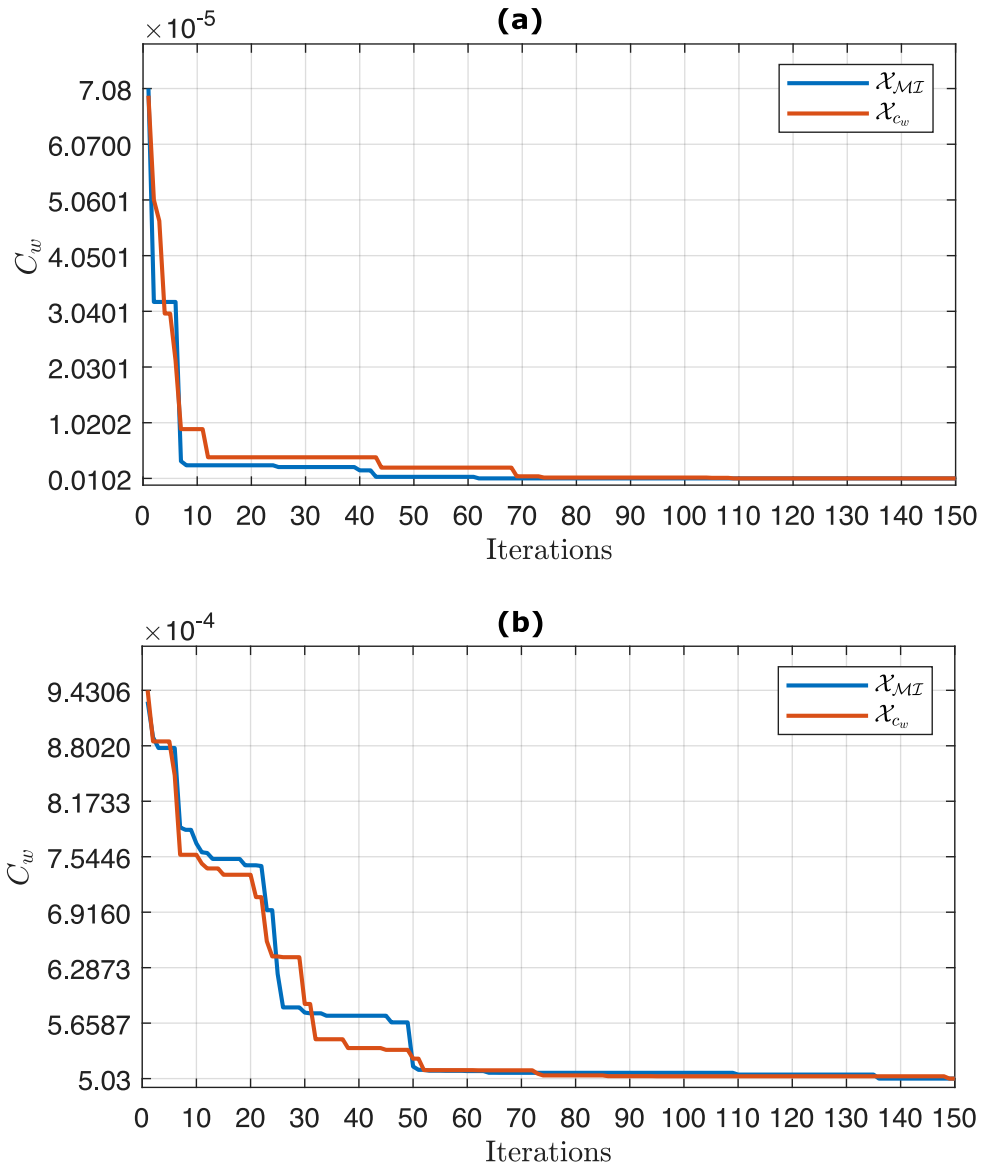


Figure 5.16: Plot showing C_w versus optimisation iterations performed to optimise (a) PD and (b) DTMB hulls in $\mathcal{X}_{\mathcal{MI}}$ and $\mathcal{X}_{c_w}$.

Table 5.8: C_w values of optimal designs obtained when optimisation is performed in $\mathcal{X}_{\mathcal{MI}}$ and $\mathcal{X}_{c_w}$ for PD and DTMB hulls.

C_w							
Runs	#1	#2	#3	#4	#5	Average	STD
Design space	PD Hull						
$\mathcal{X}_{\mathcal{MI}}$	1.0328E-07	1.0395E-07	1.0244E-07	**1.0205E-07**	1.0505E-07	1.0335E-07	1.2013e-09
$\mathcal{X}_{c_w}$	1.0304E-07	1.0318E-07	1.0355E-07	1.0303E-07	**1.0302E-07**	1.0316E-07	2.2546e-10
	DTMB Hull						
$\mathcal{X}_{\mathcal{MI}}$	5.0809E-04	5.0945E-04	5.1097E-04	**5.0314E-04**	5.1419E-04	5.0917E-04	3.6349E-06
$\mathcal{X}_{c_w}$	5.0455E-04	5.0455E-04	**5.0269E-04**	5.0616E-04	5.0440E-04	5.0447E-04	1.09891E-06

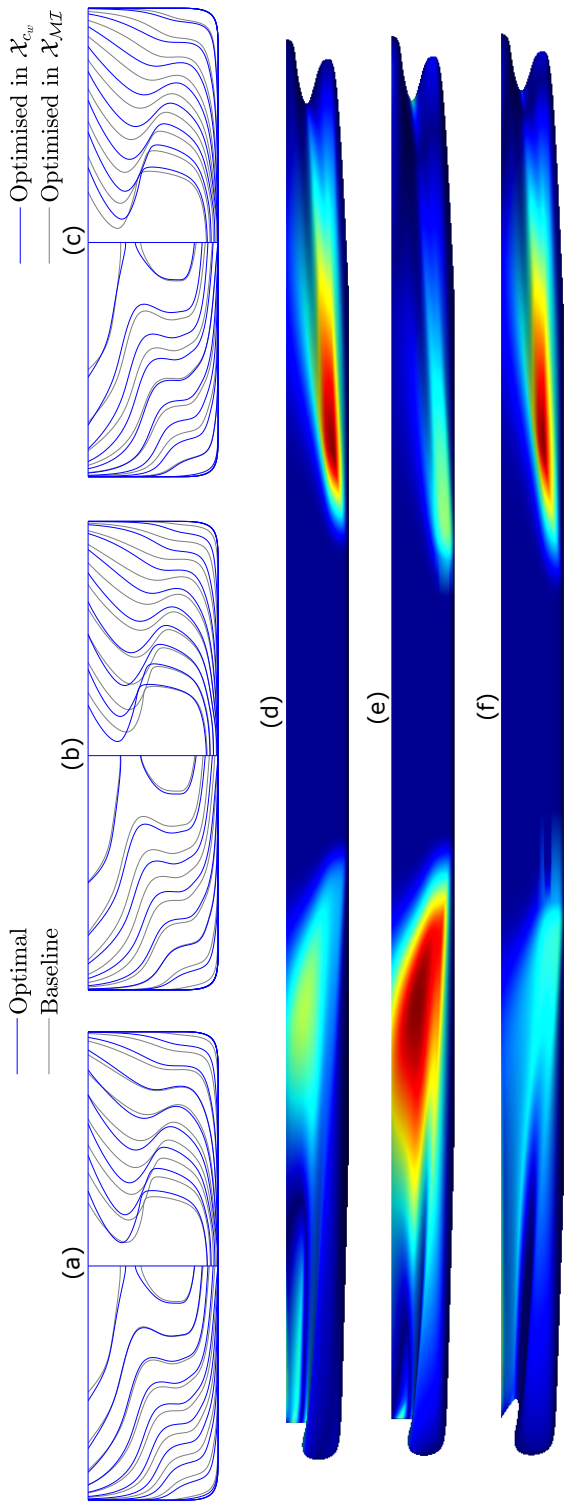


Figure 5.17: Comparison between the baseline and optimised PD hulls obtained from $\mathcal{X}_{\mathcal{MI}}/\mathcal{X}_{c_w}$ in term of (a)/(b) bodyplans and (d)/(e) one-sided Hausdorff Distance, respectively. Similar comparison between optimised designs of $\mathcal{X}_{c_w}$ and $\mathcal{X}_{\mathcal{MI}}$ in term of (c) bodyplans and (f) one-sided Hausdorff Distance.

DTMB hull, parameters sensitive to C_w and $\mathcal{MI}^4$ are $\mathbf{t}_{\mathcal{MI}} = \{t_4, t_8, t_{14}, t_{15}, t_{17}, t_{22}, t_{26}\}$ and $\mathbf{t}_{c_w} = \{t_4, t_8, t_{14}, t_{15}, t_{22}, t_{26}\}$ and again, apart from t_{17}, parameters sensitive to both sets are the same. However, geometric variation between two optimised DTMB hulls is slightly higher compare to PD hulls. However, despite the design difference both optimised designs have similar performance, e.g., their C_w is equal to 5.0269E-04 and 5.0314E-04.

Table 5.9: C_w values and the percentage improvement in baseline design of PD and DTMB hulls made when shape optimisation is performed in sensitive (($\mathcal{X}_{\mathcal{MI}}$ and $\mathcal{X}_{c_w}$)) and original ($\mathcal{X}$) design spaces.

	Baseline Design	Design from $\mathcal{X}_{c_w}$	Design from $\mathcal{X}_{\mathcal{MI}}$	Design from $\mathcal{X}$
		PD Hull		
C_w	1.0678E-04	1.0302E-07	1.0205E-07	1.05081E-07
		DTMB Hull		
C_w	1.0531E-03	5.0269E-04	5.0314E-04	8.9218E-05

The above results indicate that for both PD and DTMB hulls, the optimal designs obtained from $\mathcal{X}_{\mathcal{MI}}$ and $\mathcal{X}_{c_w}$ have similar performance in terms of the wave resistance C_w. Even with only seven sensitive parameters for the PD hull and six sensitive parameters for the DTMB hull in $\mathcal{X}_{\mathcal{MI}}$, a substantial improvement is made versus the baseline designs in terms of C_w. Table 5.9 summarises C_w values of the parent and optimal designs obtained from sensitive ($\mathcal{X}_{\mathcal{MI}}$ and $\mathcal{X}_{c_w}$) and original ($\mathcal{X}$) design spaces. This shows that the biggest improvement for the PD hull is achieved when shape optimisation is performed with sensitive parameters instead of using the entire set of 26 design space parameters. These impaired optimisation results obtained from $\mathcal{X}$ can be attributed to its high dimensionality. As $\mathcal{X}$ is 27-dimensional for the PD hull, optimisation exhibits slow convergence, requiring more design evaluations, which results in a higher computational cost. Furthermore, more than half of the improvement made to the DTMB design achieved with a full set of 27 design parameters of the DTMB hull is contributed by only six parameters sensitive to $\mathcal{MI}^4$.

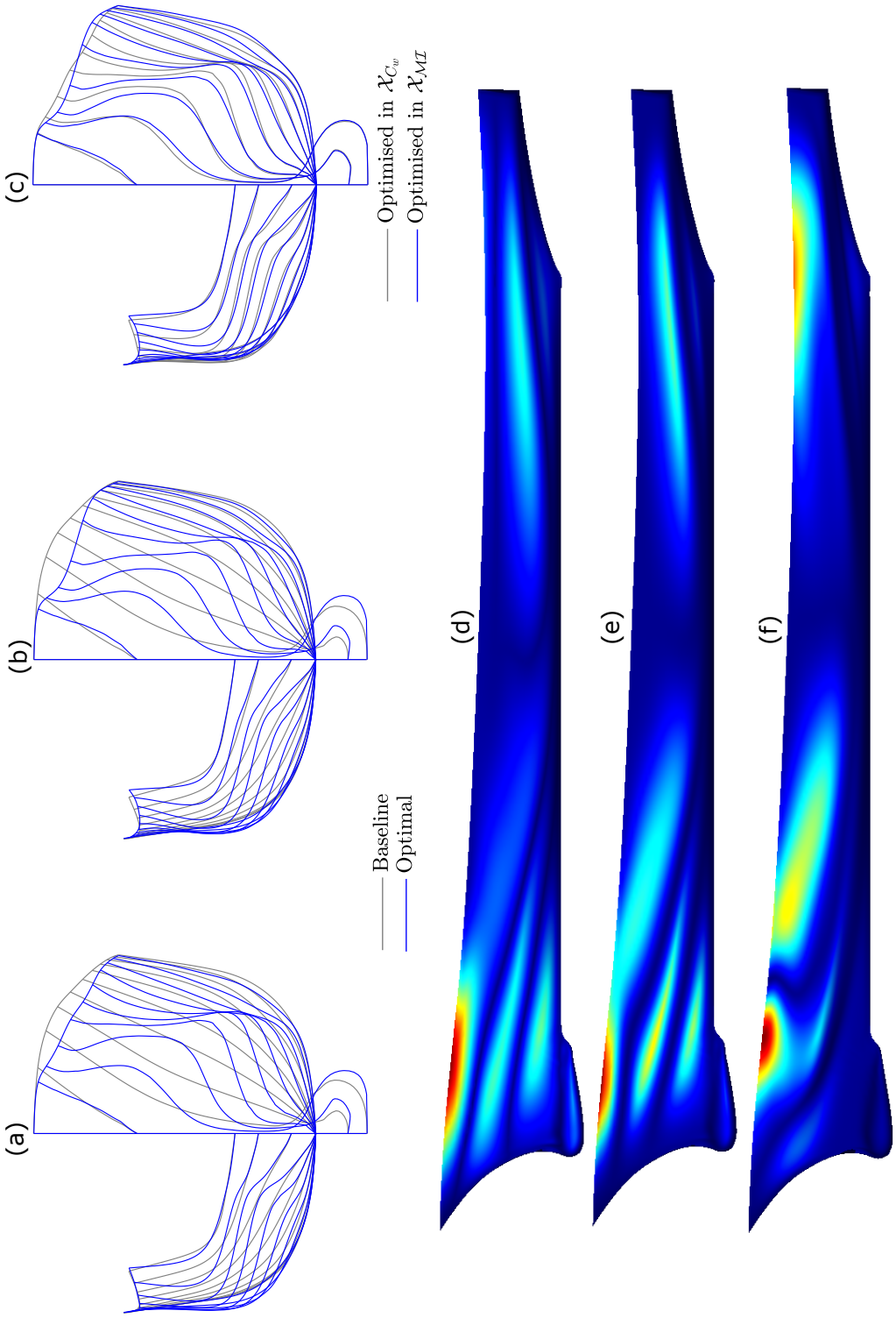


Figure 5.18: Comparison between the baseline and optimised DTMB hulls obtained from $\mathcal{X}_{\mathcal{MI}}/\mathcal{X}_{c_w}$ in term of (a)/(b) bodyplans and (d)/(e) one-sided Hausdorff Distance, respectively. Similar comparison between optimised designs of $\mathcal{X}_{c_w}$ and $\mathcal{X}_{\mathcal{MI}}$ in term of (c) bodyplans and (f) one-sided Hausdorff Distance.

5.4.9 Computational cost

The computational cost to perform SA to geometric moments is glaringly less than performing SA with $\mathcal{X}_{c_w}$ and $\mathcal{X}_{\mathcal{MI}}$. On a PC with Intel(R) Xeon(R) Gold 6226 CPU with 2.70GHz and 2.69 GHz processors and 128GB of memory on average, it takes 11.552 and 9.039 seconds to evaluate the fourth-order shape-signature vector ($\mathcal{MI}^4$) for the PD and DTMB hull geometries composed of triangulated meshes with 1,968,835 and 2,512,886 vertices, respectively. For the PD and DTMB hulls, IGA-BEM and potential flow solvers take approximately 86.505 and 69.297 seconds to evaluate C_w. Consequently, for the PD hull, the overall computational cost for performing SA with respect to global $\mathcal{MI}^4$ and C_w is 29.0154 and 216.2701 hours, respectively. Similarly, the computational cost for performing SA for DTMB hull with respect to C_w is also significantly high. Therefore, this proves that performing SA via geometric moments can provide a prior estimation of the parameters' sensitivity with extensively reduced computational cost.

5.4.10 Limitation of geometric moment invariants for sensitivity analysis

So far, in this section, we have demonstrated that SSV composed of geometric moment invariants up to fourth-order are capable of revealing parametric sensitivities for both test cases. However, despite their usefulness to expedite SA, moments can only be used for physical quantities, with tight coupling with the geometry, such as C_w. In general, an experienced designer/engineer can easily identify the dependence of a physical quantity on geometry, thereby on moments. Therefore, one can make a technically sound judgement on using geometric moments to preliminary drive the sensitivity of the parameters for their problem. Our use of geometric moments is based on the fact that, like most physical quantities, moments are sensitive to the variation of shape features, and the sensitive parameters are those with a high effect on the shape and thus on the associated physics. However, it is not unlikely that some parameters may have a high impact on the shape but a negligible impact on the physics in a design problem. In that case, one may require

SSV composed of geometric moments of higher than fourth-order. For these reasons, moments may not be able to scale well versus other physical quantities as they did in the present case, especially for black-box problems or problems with no strong dependence of physical QoI on moments, such as it is the case of frictional resistance of the hull. Thus, a good understanding of the underlying physics is necessary to perform a geometric-moment dependent SA. In addition, moments can only be used for the SA of 2D/3D design problems. In contrast, SA is also widely used in the context of numerical problems with no geometrical objects/domains like [36, 37].

Nevertheless, there exist a wide variety of problems [14,113,114,116–118,126] for which performing geometric-moment dependent SA can be very beneficial to reduce the computational cost of working directly with the physics. Moreover, as discussed in §5.3, our methodology is based on two pillars:

- The collocation BEM for Fredholm Boundary Integral Equations (BIE) of the second type, used for formulating and solving the elliptical exterior boundary-value problem (BVP) associated with the chosen QoI. Along with the Galerkin Finite-Element Method (FEM), collocation BEM provides a standard weak formulation for solving various problems in continuum mechanics and is especially suitable for BVP's defined on infinite domains, which is exactly the case for the wave-resistance problem.

- A pair of parametric modelers (PD [1] and DTMB [12]), capable to parametrise in robust and efficient manner complex free-form objects.

Based on the above remarks, we believe that the proposed approach can be applied to a broader class of shape optimisation problems that can be modelled via BIE on free-form geometries. Even if there is no strong connection of physics under consideration with geometric moments, they can at least capture the sensitivity of parameters to shape variation in a pure geometrical setting. This can be very useful at the preliminary stage design stage of constructing a parametric model, where the decision on type and dimensionality of parameterisation is made based on the effect a parameter can have on the original shape.

This is related to the previously mentioned fact that designers are interested in parameterisation at the initial design stage, which can deliver the highest variability possible. This is especially of interest for parametric generative design [8, 54].

Despite the limitations mentioned above, the results presented herein support the assertion that at least in the field of naval architecture, ocean and marine engineering, where these techniques are widely used, the proposed approach constitutes a valuable contribution. Furthermore, our end aim is to find a class of design problems specifically in maritime and aeronautical fields, where these approaches are extensively used and ultimately, to trigger an interest in the research community towards exploring such physics-correlated but computationally inexpensive quantities to perform SA as a priori for identifying the sensitivity of parameters.

5.5 Conclusion and future works

This work describes our quest to support computationally demanding physical models with the aid of efficient geometric quantities such as geometric moments and their invariants. Using such geometric quantities, we proposed a method to expedite Sensitivity Analysis (SA) in the context of shape optimisation of 3D free-form shapes such as ship hulls. Our choice of geometric moments is based on the fact that they are intrinsic properties of solid shapes' underlying geometry that can provide essential design indications to facilitate designers in CAD. The set of geometric moment invariants up to particular order can also be used to create a shape-signature vector, which approximates the shape as order increases. Moreover, computing geometric moments is also vital for physics-based simulations that help in improving realism in physical animations. To prove that geometric moments can benefit designers as a prior check on the sensitivity of parameters, we utilised wave-resistance coefficient (C_w) as a physical quantity, as it is a crucial design consideration for a ship hull design towards improving efficiency and thus decreasing Fuel Oil Consumption. The distribution of the hulls' geometry, especially longitudinally, has a similar impact

on geometric moments as C_w. To validate our claim, we utilised two different hull models, PD and DTMB hulls, which are constructed, parametrised and physically evaluated with two different approaches. Various experiments are performed with varying degrees of both global and composite shape-signature vectors and C_w to experimentally quantify the degree of similarity between the parameters sensitive to these quantities. The results from these experiments revealed a good correlation between the sensitive parameters obtained from the fourth-order composite shape-signature vectors ($\mathcal{MI}^s$) and C_w. In the case of the PD hull, seven parameters sensitive to $\mathcal{MI}^4$ are also among the 8 parameters sensitive to C_w. Interestingly, similar results are obtained for the DTMB hull, where 6 out of 7 sensitive parameters to C_w are also sensitive to $\mathcal{MI}^4$. Afterwards, two different design spaces are constructed for both hull models, one with sensitive parameters obtained with C_w and the other with $\mathcal{MI}^s$. Shape optimisation is performed in both spaces performed via a meta-heuristic optimisation approach. Final optimisation results showed that the design generated from design space constructed with sensitive parameters of C_w and $\mathcal{MI}^4$ for both types of hulls offer similar performance; however, interestingly, the optimal hull designs from $\mathcal{MI}^4$ have sightly better performance.

In future work, our prime aim is to explore other computationally demanding engineering design problems for which geometric moments can aid parametric analysis, specifically intra-sensitivity analysis [18]. Moreover, we are also interested in exploring other shape integral properties along with their usage to support surrogate and reduced-order modelling, specifically in the context of physics-informed learning [21].

Chapter 6

Shape-supervised dimension reduction

6.1 Introduction

Simulation-driven optimisation of free-form shapes is often obstructed by high-dimensional design spaces stemming from the baseline/parent shape parameterisation, which leads to the notorious curse of dimensionality [32]. A common cure involves dimensionality reduction, referred to as feature extraction/embedding or manifold learning. These approaches aim to extract latent features/variables from the design space, which can be classified as geometrically active or inactive depending on their importance in affecting a shape's geometric variability [21]. Inactive features are redundant, and their usage has no or minimal impact on shape variation and performance improvement during optimisation; thus, they can be safely ignored to reduce the space's dimensionality. The geometrically active latent features form a new set of parameters for shape modification and construct a basis spanning a lower-dimensional subspace for faster optimisation convergence with fewer computationally intensive design evaluations. The widely used Design Space Dimensionality Reduction (DSDR) approaches include the Karhunen-Loève Decomposition (KLD) [22] (closely related to Principal Component Analysis (PCA), the so-called proper

orthogonal decomposition [12,16]) and their non-linear extensions, such as kernel PCA [23], ISOMAP [24], LLE [25] to handle design space non-linearities if present. Recently, Machine Learning-based approaches, autoencoders [23, 26], Generative-Adversarial Networks (GANs) [32], and variations [27, 28], emerging from applications in image analysis, object recognition, speech analysis, clustering, and data visualisation etc., have gained attention in DSDR literature.

Despite the demonstrated efficiency of the aforementioned approaches for DSDR, they often suffer from certain drawbacks. A common deficiency is their inability to preserve a shape's complexity and intrinsic underlying geometric structure. Thus, the resulting subspace lacks the representation *capacity* and *compactness*, which, as defined in [28, 32], is subspace's ability to produce *diverse* and *valid* shapes, respectively, with least number of latent variables when being explored for shape optimisation. These deficiencies can hamper the success of the optimiser as it may spend the majority of the available computational budget on exploring infeasible, practically invalid and similar shapes. Furthermore, the basis of the subspace is solely formulated with geometric features and no information related to physics, against which designs are assessed, is incorporated. Therefore, it may not be an optimisation-efficient subspace because, even if the high geometric variation is preserved, maximum design improvements are not guaranteed; see [21, 26, 27, 33]. However, it should be noted that these techniques' inability to extract appropriate geometric or physics-associated features is not necessarily an intrinsic characteristic; it mainly stems from the geometry representations used in subspace learning, which are commonly low-level shape discretisations. Thus, extracting intrinsic latent information from such representations becomes implausible; therefore, richer representations with high-level details related to the underlying shape's structure and physics are imperative.

6.1.1 Objective and contribution

To simultaneously tackle the aforementioned challenges associated with DSDR, we propose a *shape-supervised approach*, which, with the geometric modification function [12], uses the

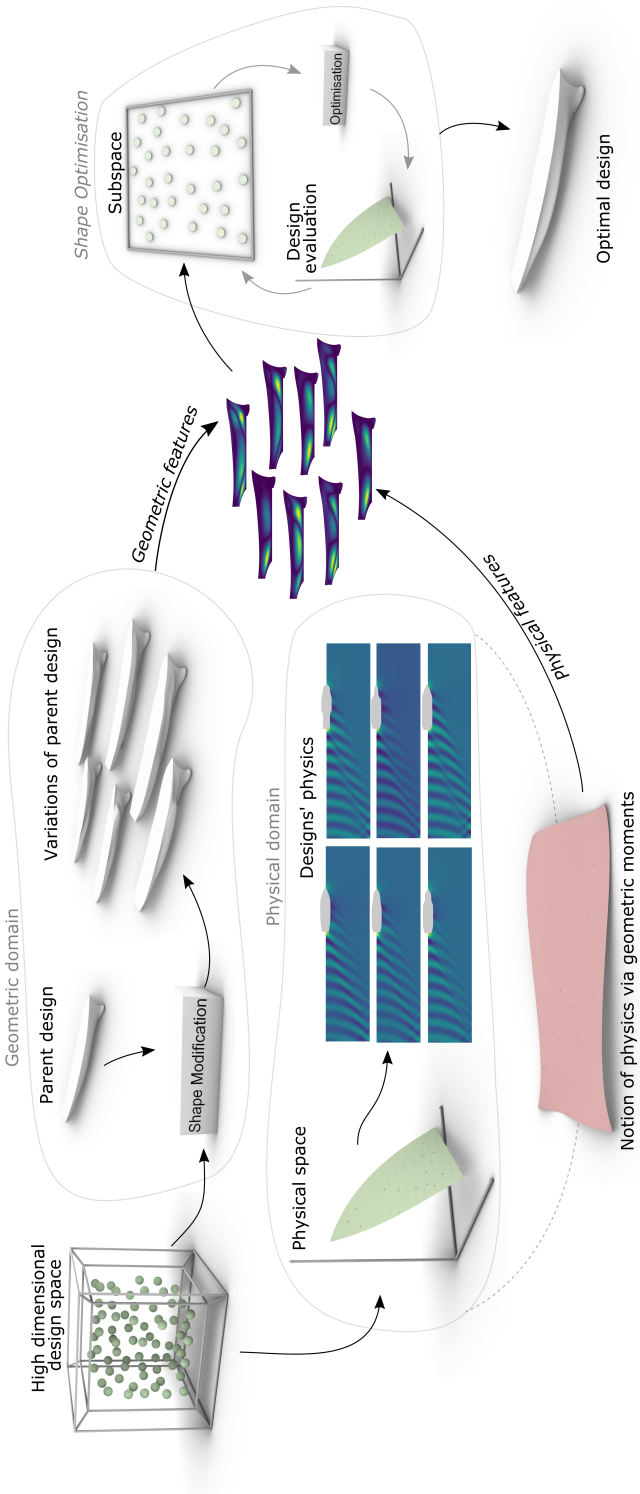


Figure 6.1: Workflow illustration of the proposed shape-supervised DSDR approach. It commences by extracting latent features from a shape's geometric and physical domains and combines them in a subspace that guarantees geometric and functional variability. For reasons of computational efficiency, geometric moment invariants are used to capture physical information. Once created, the subspace is connected to the optimiser and the design evaluation module to expedite convergence to the optimal solution.

shape's integral properties, i.e., geometric moments and their invariants [14,51], to harness the compact geometric representation of the baseline shape and complement its physics during DSDR. Therefore, the resulting subspace

1. has not only enhanced representation capacity and compactness to produce a valid and diverse set of design alternatives, respectively, but
2. is also physically informed to improve the convergence rate of the shape optimiser towards an optimal solution.

As mentioned in the previous chapter, a well-known feature of geometric moments is their strong coupling with physics as they provide the geometric foundation for different physical analyses and, like physics, provide important clues about the form, distribution and validity of the design. As with physics, they depend on the design's geometry, but their evaluation is substantially less expensive. Therefore, we perform DSDR in a supervised setting where geometric moments are used as QoI to induce a notion of the shape's physical information, thus resulting in a shape-supervised subspace whose basis is not only associated with the shape's geometry but also with its physics. A complete workflow of the proposed pipeline is illustrated in Fig. 6.1. In conclusion, the selection of geometric moments in our work is motivated by the following fundamental insights:

1. Geometric moments of a shape are the intrinsic properties of its underlying geometry and act as a unifying medium between geometry and its physical evaluation [14, 51].
2. The analysis for any physics requires such integral properties of the geometry. At the same time, moments of a domain are sufficient to ensure accurate integration of a large class of integrands [14, 113].
3. Like physics, geometric moments also act as a compact shape signature or descriptor facilitating various shape processing tasks [116, 118].

Geometric moments of higher order are also used in different shape processing tasks such as object recognition [118], rigid body transformation [116], etc. Most notably, in physical analyses, they are used for parametric sensitivity analyses [150], material field modelling [114], governing equations of motions for flows around a body [113], and, recently, for meshfree FEA [14, 127], in which moment-based shape representations are used to aid the interoperability between CAD representations and physics.

6.1.2 Overview of the proposed approach

To maximise the accumulation of both geometric and physical variance in the subspace, our approach uses a set of composite moments by disintegrating the body geometry into several subsets of coherent shape. Afterwards, we use the divergence theorem to evaluate moments of all subsets up to a specific order. Once moments are evaluated, they are used, along with the shape modification function, to form a Shape Signature Vector (SSV) function, which acts as a descriptor to represent each instance in the design space uniquely. Karhunen-Loève Expansion (KLE) of SSV is evaluated, where the solution of a variational problem allows for the evaluation of latent features as a linear combination of original designs. The features provided by KLE are expressed by the eigenfunctions of a symmetric and positive definite covariance function constructed with SSV. The KL-values associated with each feature allow the identification of active and inactive features. The active features reparameterise the shape and act as a new basis to span the subspace, retaining the highest variance in geometry and physics. Moreover, different quality measures are formulated to assess the quality of the shape-supervised subspace in producing rich and valid sets of shapes.

Two test cases, a three-dimensional (3D) wing and a ship hull, are used to analyse the shape-supervised subspace's ability to produce diverse and valid designs. At the same time, the ship hull model is also used for shape optimisation with respect to the wave resistance coefficient (C_w). C_w is a significant component of the total ship's resistance and a critical design criterion whose evaluation is computationally demanding. These

experiments validate the conservation of physical information via geometric moments and expedite convergence to optimal solutions.

6.1.3 Related works

Recently, GANs [32] have been used for reparameterisation of the shape with latent features, as baseline parameterisation approaches, like the ones used in free-form deformation, produce high-dimensional design spaces [18] that do not guarantee feasible/valid shapes. Chen et al. [32] proposed the Bézier-GAN for two-dimensional (2D) aerofoil design by introducing a Bézier layer into GAN to maximise subspace's representation capacity and compactness. However, the baseline parameterisation cannot automatically guarantee valid shapes and, as a result, these approaches require a training dataset of *existing designs*, which may prevent optimisers from finding innovative designs; a drawback studied in detail by Li and Zhang in [28]. Moreover, their usage can be problematic for novel problems, as, in this case, creating training datasets can be extremely arduous.

Furthermore, although subspaces resulting from the approaches described above may address the validity problem, physics-associated features are still not present. The Active Subspace Method [79] and supervised KLE [33] can handle this issue, but they become computationally intensive as they require direct evaluation of physics quantities and/or their gradients. Yonekura and Suzuki [26] recently used a conditional variational auto-encoder (CVAE) for aerofoil design. They used the lift coefficient as a condition to auto-encoder during training so that the decoder could generate the shape with specific performance. Chen and Ahmed [96] proposed PaDGAN to augment the design performance into the generator to create high-quality designs with good optimisation convergence. Another GANs-based method was proposed by Shu et al. [27], which elevates the quality of generated designs by iteratively updating the training dataset using performance-based design filtering. All methods above, i.e., [26, 27, 33, 79, 96] are efficient but supervised and therefore require performance labels to be evaluated for a large shapes dataset; if not readily available, the creation of such a dataset is computationally very demanding.

The remainder of this chapter is organised as follows: Section 6.2 discusses the problem formulation, evaluation of geometric moments and KLE of SSV. A detailed discussion on the relevance of geometric moments with C_w, along with the test case description, is given in Section 6.3. The numerical results of the proposed technique are provided in Section 6.4. Section 6.5 presents concluding remarks and opportunities for future work.

6.2 Proposed approach

This section provides an in-depth description of the proposed approach, including the general assumptions and the mathematical formulation of SSV generation. A brief overview of the criteria used to assess the quality of a subspace is also provided.

6.2.1 Problem formulation

Let Γ be a 3D body bounded by a closed 2D manifold $\mathcal{G}$, representing a baseline/parent design, and $\bar{\boldsymbol{\vartheta}} \in \mathcal{G} \subseteq \mathbb{R}^{\bar{n}}$, with $\bar{n} = 1, 2, 3$, a coordinate set on this manifold. For an automatic shape modification, $\mathcal{G}$ is commonly parameterised with n geometric parameters, defining the parametric/design vector $\mathbf{t} = (t_1, t_2, \ldots, t_n) \in \mathcal{T} \subseteq \mathbb{R}^n$. Here, $\mathcal{T}$ is the $n-$dimensional design space, which is bounded by appropriately defined set constraints, e.g., $\mathcal{T} := \left\{\mathbf{t} : t_i^l \leq t_i \leq t_i^u, \forall i \in \{1, 2, \ldots n\}\right\}$ with $\mathbf{t}^l, \mathbf{t}^u \in \mathbb{R}^n$ denoting the lower and upper bound vector, respectively. The parametric vector $\mathbf{t}$ of $\mathcal{G}$ yields a continuous shape modification vector $\boldsymbol{G}(\bar{\boldsymbol{\vartheta}}, \mathbf{t}) \in \mathbb{R}^{n_G}$ with $n_G = 1, 2, 3$, which for any $\mathbf{t} \in \mathcal{T}$ modifies the initial $\bar{\boldsymbol{\vartheta}}$ to produce new $\bar{\boldsymbol{\vartheta}}'$ that defines the modified $\mathcal{G}'$, i.e.,

$$\bar{\boldsymbol{\vartheta}}' = \bar{\boldsymbol{\vartheta}} + \boldsymbol{G}(\bar{\boldsymbol{\vartheta}}, \mathbf{t}), \forall \bar{\boldsymbol{\vartheta}}. \tag{6.1}$$

Furthermore, in shape optimisation, we also assume at least one given function $g : \mathcal{T} \to \mathbb{R}$ which, for each $\mathbf{t} \in \mathcal{T}$, evaluates the performance index $\ell = -g(\mathbf{t}) \in \mathbb{R}$ of the corresponding design. Therefore, the optimisation problem can be stated as follows:

$$\text{Find}\, \mathbf{t}^* \in \mathbb{R}^n : -\ell^* = g(\mathbf{t}^*) = \min_{\mathbf{t}^* \in \mathcal{T}} g(\mathbf{t}). \tag{6.2}$$

Obviously, the optimal design will be then defined by the corresponding set $\bar{\boldsymbol{\vartheta}}^*$, with $\bar{\boldsymbol{\vartheta}}^* = \bar{\boldsymbol{\vartheta}} + \boldsymbol{G}(\bar{\boldsymbol{\vartheta}}, \mathbf{t}^*)$.

In a typical shape optimisation problem, we may use a set of alternative performance indices (multi-objective optimisation), and additional functional constraints (design specifications/requirements) expressed as inequalities and equalities that further limit the space of feasible designs. Finally, an appropriate optimisation method is employed to search for the optimum solution ($\mathbf{t}^*$) within the feasible space bounded by all imposed constraints.

6.2.2 Design space dimensionality reduction

As explained in §6.1, the computational cost of shape optimisation increases exponentially with the dimension of $\mathcal{T}$. This cost grows further if evaluating the performance index ℓ is complicated and time-consuming. Therefore, in the present work, we intend to cure the curse of dimensionality with feature extraction techniques to create a lower-dimensional subspace using DSDR. Typically, dimensionality reduction is achieved via extraction of latent features/variables of $\mathcal{T}$, which reduces its dimension while retaining, to the extent possible, the geometric variability exhibited in the resulting domains $\mathcal{G}'$. However, due to the aforementioned drawbacks associated with a typical DSDR approach, we aim to develop a subspace with latent variables that go beyond the features extracted from $\mathcal{T}$. These additional elements comprise appropriate geometric moments computed on the body geometry. Therefore, the resulting subspace is both adequately rich and robust and efficient when used for shape optimisation, as we will demonstrate in §3.

To construct this subspace, we consider that along with the continuous shape modification vectors, $\boldsymbol{G}(\bar{\boldsymbol{\vartheta}}, \mathbf{t})$, there is a lumped geometric moment vector, $\boldsymbol{M}(\boldsymbol{\vartheta}_M, \mathbf{t}) \in \mathbb{R}^{n_M}$ with $n_M = 1, 2, \ldots$, which has a null measure and corresponds to an arbitrary point, $\boldsymbol{\vartheta}_M$, where this moment vector is virtually defined. We further assume $\mathcal{G}$ and $\mathcal{M}$ as domains

of definition for $\boldsymbol{G}(\bar{\boldsymbol{\vartheta}}, \mathbf{t})$ and $\boldsymbol{M}(\boldsymbol{\vartheta}_M, \mathbf{t})$, respectively; see Fig. 6.2. Now, consider a combined geometry and moment vector $\boldsymbol{P}(\boldsymbol{\vartheta}, \mathbf{t}) \in \mathbb{R}^{n_P}$, $n_P = n_G + n_M$, defined in the domain $\mathcal{P} := \mathcal{G} \cup \mathcal{M}$ with $\boldsymbol{\vartheta} = (\bar{\boldsymbol{\vartheta}}, \boldsymbol{\vartheta}_M)$ and

$$\boldsymbol{P}(\boldsymbol{\vartheta}, \mathbf{t}) = \left(\boldsymbol{G}(\bar{\boldsymbol{\vartheta}}, \mathbf{t}), \boldsymbol{M}(\boldsymbol{\vartheta}_{\boldsymbol{M}}, \mathbf{t})\right). \tag{6.3}$$

$\boldsymbol{P}(\boldsymbol{\vartheta}, \mathbf{t})$ contains both the geometry and its moments and forms a unique SSV function encompassing high-level information about the baseline design. Also consider that $\boldsymbol{P}(\boldsymbol{\vartheta}, \mathbf{t})$ belongs to a disjoint Hilbert space $L_f^2(\mathcal{P})$ as shown in Fig. 6.2, which is defined by the generalised inner product:

$$\begin{aligned} (\mathbf{a}, \mathbf{b})_f &= \int_{\mathcal{P}} f(\boldsymbol{\vartheta}) \mathbf{a}(\boldsymbol{\vartheta}) \cdot \mathbf{b}(\boldsymbol{\vartheta}) \mathrm{d}\boldsymbol{\vartheta} \\ &= \int_{\mathcal{G}} f(\bar{\boldsymbol{\vartheta}}) \mathbf{a}(\bar{\boldsymbol{\vartheta}}) \cdot \mathbf{b}(\bar{\boldsymbol{\vartheta}}) \mathrm{d}\bar{\boldsymbol{\vartheta}} + f(\boldsymbol{\vartheta}_M) \mathbf{a}(\boldsymbol{\vartheta}_M) \cdot \mathbf{b}(\boldsymbol{\vartheta}_M), \end{aligned} \tag{6.4}$$

with the associated norm $\|\mathbf{a}\| = (\mathbf{a}, \mathbf{a})_f^{\frac{1}{2}}$, where $f(\bar{\boldsymbol{\vartheta}}), f(\boldsymbol{\vartheta}_M) \in \mathbb{R}$ are appropriate positive weight functions used to focus analysis on certain regions of $\mathcal{G}$.

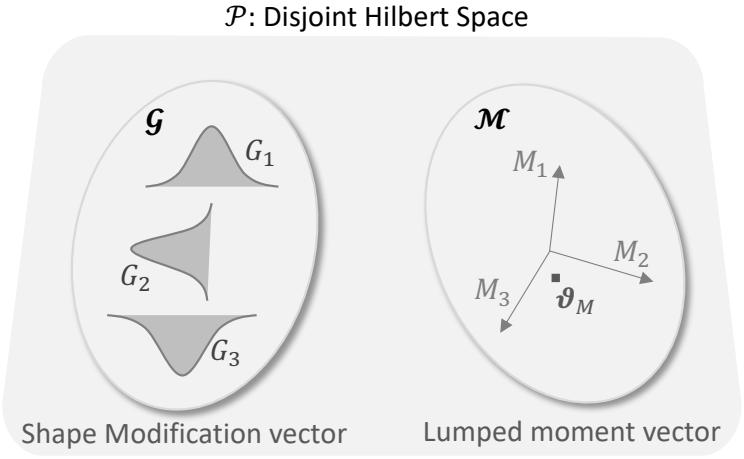


Figure 6.2: Domains for shape modification vector and lumped geometric moment vector in a disjoint Hilbert space.

The identification of optimal design through this process may suffer from epistemic uncertainties [22]. Therefore, one can consider $\mathbf{t}$ as an element of a stochastic space $\mathcal{T}$ with an associated Probability Density Function (PDF) $\rho(\mathbf{t})$, which represents the prior probability of finding optimal design in a given $\mathcal{T}$. An appropriate definition of $\rho(\mathbf{t})$ is nontrivial; therefore, the prior is usually defined as a uniform distribution function, i.e., any realisation of $\mathbf{t}$ has the same probability of being $\mathbf{t}^*$. Once $\rho(\mathbf{t})$ is defined, the mean and the variance of SSV can be evaluated as

$$\langle \boldsymbol{P} \rangle = \int_{\mathcal{T}} f(\boldsymbol{\vartheta}) \boldsymbol{P}(\boldsymbol{\vartheta}, \mathbf{t}) \rho(\mathbf{t}) \mathrm{d}\mathbf{t}, \tag{6.5}$$

$$\sigma^2 = \langle \|\overline{\boldsymbol{P}}\|^2 \rangle = \int_{\mathcal{T}} \int_{\mathcal{P}} f(\boldsymbol{\vartheta}) \overline{\boldsymbol{P}}(\boldsymbol{\vartheta}, \mathbf{t}) \cdot \overline{\boldsymbol{P}}(\boldsymbol{\vartheta}, \mathbf{t}) \rho(\mathbf{t}) \mathrm{d}\boldsymbol{\vartheta} \mathrm{d}\mathbf{t}, \tag{6.6}$$

where $\overline{\boldsymbol{P}}$ is the *deviation from the mean of SSV* (i.e., $\overline{\boldsymbol{P}} = \boldsymbol{P} - \langle \boldsymbol{P} \rangle$) and $\langle \cdot \rangle$ is the ensemble average over $\mathbf{t}$. The aim for dimensionality reduction is to find the lower-dimensional representation of $\overline{\boldsymbol{P}}(\boldsymbol{\vartheta}, \mathbf{t})$, namely, $\overline{\boldsymbol{P}}(\boldsymbol{\vartheta}, \mathbf{v})$, which, instead of $\mathbf{t}$ depends on a *Geometrically- and Functionally-Active Latent Variable* (GFALV) vector, $\boldsymbol{v} = \{v_1, v_2, v_3, \ldots, v_m\} \in \mathcal{V} \subseteq \mathbb{R}^m$. GFALV is constructed using an appropriate combination of features from $\mathcal{T}$ and SSV which will constitute the coordinates in a new m-dimensional subspace, $\mathcal{V} := \left\{ \boldsymbol{v} : v_i^l \leq v_i \leq v_i^u, \forall i \in \{1, 2, \ldots m\} \right\}$ with $m < n$, i.e., $\mathcal{V}$ is a low-dimensional space when compared with the original design space, $\mathcal{T}$. This new vector space can be employed to expedite shape optimisation. Fig. 6.3 graphically illustrates the notions of shape modification via the original and proposed approaches.

In the construction of SSV, introduced in Eq. (6.3), we use a finite number of moments of Γ, which are defined as in Eq. (5.2) and evaluated as described previously in §5.2.2.

6.2.3 Karhunen-Loève expansion of SSV

After the initial construction of SSV with the invariant geometric moments, we employ KLE, which aims to find an optimal basis of orthonormal functions for the linear repre-

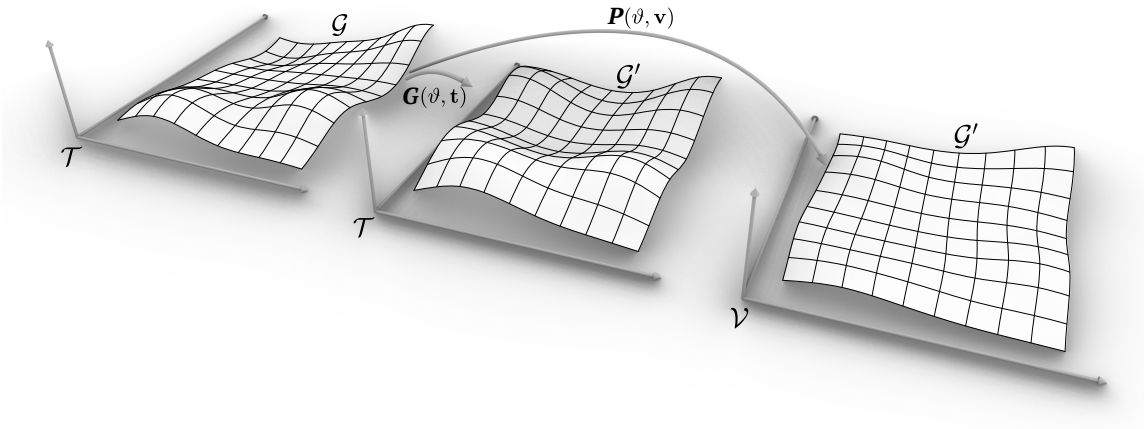


Figure 6.3: Representation of the scheme and notation used for the current formulation of shape modification.

sentation of SSV so that:

$$\overline{\boldsymbol{P}}(\boldsymbol{\vartheta}, \mathbf{t}) \approx \sum_{i=1}^{m} v_i \boldsymbol{\omega}_i(\boldsymbol{\vartheta}), \tag{6.7}$$

where $\left\{\boldsymbol{\omega}_i(\boldsymbol{\vartheta})\right\}_{i=1}^{m}$ are orthonormal functions forming the basis of the subspace $\mathcal{V}$ which will retain, to the extent possible, the variance in shapes and moments exhibited in $\mathcal{P}$. These functions are used to form the GFALV vector, $\boldsymbol{v} \in \mathcal{V}$, whose ith element can be represented as

$$v_i = \left(\overline{\boldsymbol{P}}, \boldsymbol{\omega}_i\right)_f = \int_{\mathcal{P}} f(\boldsymbol{\vartheta}) \overline{\boldsymbol{P}}(\boldsymbol{\vartheta}, \mathbf{t}) \cdot \boldsymbol{\omega}_i(\boldsymbol{\vartheta}) \mathrm{d}\boldsymbol{\vartheta}, \tag{6.8}$$

which, as explained earlier, will be used for shape modification during optimisation. The optimal condition for KLE is to construct basis functions retaining maximum geometric variance (σ^2) via Eq. (6.7). Therefore, combining Eq. (6.6), (6.7) and (6.8) we find:

$$\begin{aligned}\sigma^2 &= \sum_{i=1}^{\infty}\sum_{j=1}^{\infty}\langle v_i v_j\rangle(\boldsymbol{\omega}_i(\boldsymbol{\vartheta}),\boldsymbol{\omega}_i(\boldsymbol{\vartheta}))_f \\ &= \sum_{j=1}^{\infty}\left\langle v_j^2\right\rangle = \sum_{j=1}^{\infty}\left\langle\left(\overline{\boldsymbol{P}},\boldsymbol{\omega}_i(\boldsymbol{\vartheta})\right)_f^2\right\rangle.\end{aligned} \tag{6.9}$$

The basis retaining the maximum variance is provided by the solution of the following variational problem [22]:

$$\begin{aligned}\min_{\boldsymbol{\omega}\in L_f^2(\mathcal{P})} \quad & J\left(\boldsymbol{\omega}(\boldsymbol{\vartheta})\right) = \left\langle\left(\overline{\boldsymbol{P}},\boldsymbol{\omega}(\boldsymbol{\vartheta})\right)_f^2\right\rangle \\ \text{subject to} \quad & \left(\boldsymbol{\omega}(\boldsymbol{\vartheta}),\boldsymbol{\omega}(\boldsymbol{\vartheta})\right)_f^2 = 1,\end{aligned} \tag{6.10}$$

which, as proven in [22], yields

$$\mathscr{L}\boldsymbol{\omega}(\boldsymbol{\vartheta}) = \int_{\mathcal{P}} f(\boldsymbol{\theta})\left\langle\overline{\boldsymbol{P}}(\boldsymbol{\vartheta},\mathbf{t})\otimes\overline{\boldsymbol{P}}(\boldsymbol{\theta},\mathbf{t})\right\rangle\boldsymbol{\omega}(\boldsymbol{\theta})\mathrm{d}\boldsymbol{\theta} = \lambda\boldsymbol{\omega}(\boldsymbol{\vartheta}), \tag{6.11}$$

where $\otimes$ is the outer product, $\boldsymbol{\theta},\boldsymbol{\vartheta}\in\mathcal{G}$, and $\mathscr{L}$ is the self-adjoint integral operator whose eigensolutions form the basis function for the linear representation of $\overline{\boldsymbol{P}}(\boldsymbol{\theta},\mathbf{t})$ given in Eq. (6.7). The resulting eigenvectors, or KL-modes $\left\{\boldsymbol{\omega}_i(\boldsymbol{\vartheta})\right\}_{i=1}^{\infty}$, are orthogonal and constitute a complete basis for $L_f^2\left(\mathcal{G}\cup\mathcal{M}\right)$. Additionally, the eigenvalues or KL-values $\{\lambda_i\}_{i=1}^{\infty}$ represent the variance,

$$\sigma^2 = \sum_{i=1}^{\infty}\lambda_i, \tag{6.12}$$

retained by the associated basis. The first m eigenvectors, i.e., $\left\{\boldsymbol{\omega}_i(\boldsymbol{\vartheta})\right\}_{i=1}^{m}$ constitute the optimal basis for the approximation in Eq. (6.7). Moreover, considering ε as the desired level of confidence for capturing the variance, m in Eq. (6.7) can be selected to satisfy

$$\sum_{i=1}^{m}\lambda_i \geq \varepsilon\sum_{i=1}^{\infty}\lambda_i = \varepsilon\sigma^2 \tag{6.13}$$

with $0<\varepsilon\leq 1$ and $\lambda_i\geq\lambda_{i+1}$.

The numerical implementation of Eq. (6.11) – or its generalised form; see Eq. (6.4) – is performed using the approach of Diez et al. in [22]. Specifically, the steps that need to be followed are presented below:

1. Define an orthonormal basis of $\mathbb{R}^{n_P}$, $\{\mathbf{e}_k\}_{k=1}^{n_P}$;

2. Express the deviation from SSV mean, $\overline{\boldsymbol{P}}$, and KL-modes $\boldsymbol{\omega}$ in term of the basis, i.e.,

$$\overline{\boldsymbol{P}}(\boldsymbol{\vartheta}, \mathbf{t}) = \sum_{k=1}^{n_P} \overline{P}_k(\boldsymbol{\vartheta}, \mathbf{t})\ \mathbf{e}_k; \quad \boldsymbol{\omega}(\boldsymbol{\vartheta}) = \sum_{k=1}^{n_P} \omega_k(\boldsymbol{\vartheta})\ \mathbf{e}_k, \tag{6.14}$$

   where $\overline{P}_k = \overline{\boldsymbol{P}} \cdot \mathbf{e}_k$, $\omega_k = \boldsymbol{\omega} \cdot \mathbf{e}_k$ and $n_P = n_G + n_M$. Note that $\overline{P}$ in Eq. (6.14) represents a realisation of SSV, associated to $\mathbf{t}$ before dimension reduction is applied. Regardless of the shape modification function, $\boldsymbol{\omega}$ in Eq. (6.14) is the solution used to form the reduced-dimensional basis for shape optimisation;

3. Compute the integral in Eq. (6.4) by discretising the domain of integration, $\bar{\boldsymbol{\vartheta}} \in \mathcal{G}$, into E quadrilateral mesh elements with measure equal to $\Delta\mathcal{G}_i$ and centroid at $\left\{\bar{\boldsymbol{\vartheta}}_i, i = 1, 2, \ldots, E\right\}$;

4. Use the spatial discretisation $\mathbf{d}(\mathbf{t})$ and $\mathbf{W}$ of $\overline{\boldsymbol{P}}(\boldsymbol{\vartheta}, \mathbf{t})$ and $\boldsymbol{\omega}(\boldsymbol{\vartheta})$, respectively;

5. Finally, recast the problem as an eigenproblem of a matrix ($\mathbf{A}$):

$$\mathbf{AW} = \mathbf{W}\boldsymbol{\Lambda}, \tag{6.15}$$

where, $\mathbf{W} = \left\{\mathbf{w}^i, i = 1, 2, \ldots, n_G E + n_M\right\}$ is a square matrix whose ith column, $\mathbf{w}^i$, is the corresponding eigenvector or KL-mode. The KL-values, $\boldsymbol{\Lambda} = \{\lambda_i, i = 1, 2, \ldots, n_G E + n_M\}$, represent the variance retained by the associated KL-mode. For example, at $n_P = 4$ (with $n_G = 3$ and $n_M = 1$), $\mathbf{A}$ can be represented as

$$\mathbf{A} = \begin{bmatrix} \mathbf{C}_{11} & \mathbf{C}_{12} & \mathbf{C}_{13} & \mathbf{C}_{14} \\ \mathbf{C}_{12} & \mathbf{C}_{22} & \mathbf{C}_{23} & \mathbf{C}_{24} \\ \mathbf{C}_{13} & \mathbf{C}_{32} & \mathbf{C}_{33} & \mathbf{C}_{34} \\ \mathbf{C}_{14} & \mathbf{C}_{24} & \mathbf{C}_{34} & \mathbf{C}_{44} \end{bmatrix} \begin{bmatrix} \mathbf{Q} & 0 & 0 & 0 \\ 0 & \mathbf{Q} & 0 & 0 \\ 0 & 0 & \mathbf{Q} & 0 \\ 0 & 0 & 0 & \mathbf{Q} \end{bmatrix}, \tag{6.16}$$

where $\mathbf{C}_{lk} = \left\langle \mathrm{d}_l(\mathbf{t}) \ \left[\mathrm{d}_k(\mathbf{t})\right]^T \right\rangle$, $\forall\ l,k = 1,2,\ldots,n_P$ and $\mathbf{Q}$ is the weighted matrix to normalise $\mathbf{C}_{lk}$, so all of its components have same influence while computing $\mathbf{A}$. For dimensionality reduction we first rearrange KL-values in $\mathbf{\Lambda}$ in descending order, i.e., $\lambda_i \geq \lambda_{i+1}$. Afterwards, we select the first m KL-values $\{\lambda_i\}_{i=1}^m$ via Eq. (6.13) along with their associated KL-modes $\left\{\mathbf{w}^i\right\}_{i=1}^m$, which correspond to features with the greatest impact on geometry changes. The spatial discretisation of $\overline{\boldsymbol{P}}(\boldsymbol{\vartheta},\mathbf{t})$ and $\boldsymbol{\omega}(\boldsymbol{\vartheta})$ (namely $\mathbf{d}(\mathbf{t})$ and $\mathbf{W}$) can now be approximated and defined as

$$\mathbf{d}(\mathbf{t}) = \begin{Bmatrix} \overline{P}_1(\bar{\boldsymbol{\vartheta}}_1,\mathbf{t}) \\ \vdots \\ \overline{P}_1(\bar{\boldsymbol{\vartheta}}_E,\mathbf{t}) \\ \overline{P}_2(\bar{\boldsymbol{\vartheta}}_1,\mathbf{t}) \\ \vdots \\ \overline{P}_2(\bar{\boldsymbol{\vartheta}}_E,\mathbf{t}) \\ \overline{P}_3(\bar{\boldsymbol{\vartheta}}_1,\mathbf{t}) \\ \vdots \\ \overline{P}_3(\bar{\boldsymbol{\vartheta}}_E,\mathbf{t}) \\ \overline{P}_1(\boldsymbol{\vartheta}_M,\mathbf{t}) \end{Bmatrix} \approx \sum_{i=1}^{m} v_i \mathbf{w}^i; \quad \mathbf{w}^i = \begin{Bmatrix} \omega_1(\bar{\boldsymbol{\vartheta}}_1) \\ \vdots \\ \omega_1(\bar{\boldsymbol{\vartheta}}_E) \\ \omega_2(\bar{\boldsymbol{\vartheta}}_1) \\ \vdots \\ \omega_2(\bar{\boldsymbol{\vartheta}}_E) \\ \omega_3(\bar{\boldsymbol{\vartheta}}_1) \\ \vdots \\ \omega_3(\bar{\boldsymbol{\vartheta}}_E) \\ \omega_1(\boldsymbol{\vartheta}_M) \end{Bmatrix}. \tag{6.17}$$

The latent variables $\boldsymbol{v} \in \mathbb{R}^m$ formulated in Eq. (6.8) can be finally obtained in a discretised form as

$$v_i = \mathbf{d}(\mathbf{t})^T \begin{bmatrix} \mathbf{Q} & 0 & 0 & 0 \\ 0 & \mathbf{Q} & 0 & 0 \\ 0 & 0 & \mathbf{Q} & 0 \\ 0 & 0 & 0 & \mathbf{Q} \end{bmatrix} \mathbf{w}^i. \tag{6.18}$$

It should be noted that the KL-modes are formulated while taking into account both geometry and geometric moments to preserve the underlying structure of $\mathcal{G}$ and to accumulate the functional information of designs in $\mathcal{T}$. Therefore, by using only the first $n_G E$ elements of column vector $\mathbf{w}^i$ in Eq. (6.18), one could form the latent variable vector which is used for the shape modification of $\mathcal{G}$ during the shape optimisation performed in the subspace $\mathcal{V} := \left\{ \boldsymbol{v} : v_i^l \leq v_i \leq v_i^u, \forall i \in \{1, 2, \dots m\} \right\}$.

6.2.4 Additional design space considerations

Apart from the dimension of the design space, the use of meaningful parameter bounds, $[\mathbf{t}^l, \mathbf{t}^u]$, is also crucial since they define the allowable/feasible domain for exploration and identification of any optimum regions or points. Generally, exploration of a large space is favoured, though it considerably increases the chances of encountering invalid and impractical designs. Although such designs can be avoided by adding more design constraints, this will inevitably make the optimisation problem in Eq. (6.2) more challenging and time-consuming. On the other hand, a narrow design space weakens the need for additional constraints but, at the same time, may eliminate large regions where highly-improved or optimum designs lie. Therefore, designers tend to use their field experience to define a design space that balances robustness and allows diversity in $\mathcal{G}$.

Bounds on subspaces

Setting the subspace's parameter bounds, $(\boldsymbol{v}^l, \boldsymbol{v}^u)$, can be even more challenging as designers have to work with latent variables, $\boldsymbol{v}$, instead of the original design variables $\mathbf{t}$. Commonly, design variables $\mathbf{t}$ have physical meaning, i.e., lengths, radii, angles, etc., whereas

any components generally expect no physical interpretation of $\boldsymbol{v}$. We need to be very cautious when setting the bounds of $\mathcal{V}$ since we have to ensure that any design produced in $\mathcal{V}$ should also be a member of the appropriately bounded $\mathcal{T}$, i.e., conforming to all design constraints and requirements. To overcome this problem, one may project the bounds of the original design space on the subspace, as illustrated in Fig. 6.4. In this setting, the range of the i-th latent parameter v_i can be evaluated as

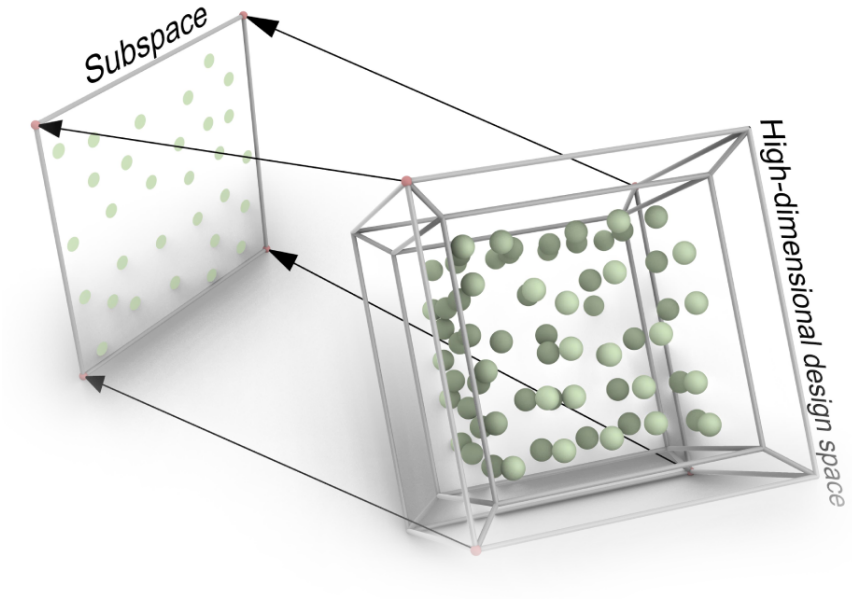


Figure 6.4: Illustration of setting the bounding limits of subspace using Eq. (6.19).

$$v_i \in \left[\min_{\mathbf{t}_\psi \in \mathcal{X}} \left(\mathbf{d}(\mathbf{t}_\psi)^T \begin{bmatrix} \mathbf{Q} & 0 & 0 \\ 0 & \mathbf{Q} & 0 \\ 0 & 0 & \mathbf{Q} \end{bmatrix} \mathbf{w}^i \right), \max_{\mathbf{t}_\psi \in \mathcal{X}} \left(\mathbf{d}(\mathbf{t}_\psi)^T \begin{bmatrix} \mathbf{Q} & 0 & 0 \\ 0 & \mathbf{Q} & 0 \\ 0 & 0 & \mathbf{Q} \end{bmatrix} \mathbf{w}^i \right) \right], \quad \psi = 1, 2, 3, \ldots, \Psi, \tag{6.19}$$

where Ψ is the number of points densely sampled from $\mathcal{T}$. Another common approach employs the standard deviation from the mean shape lying at the centroid of the design space. In this approach, the bounds for the i-th variable are set as

$$v_i \in \left[-\sqrt{\kappa\lambda_i}, \sqrt{\kappa\lambda_i}\right], \quad \kappa \in \{1, 2, 3\}. \tag{6.20}$$

The latter approach is computationally efficient, and our experiments have shown that it can provide a good balance between the number of invalid shapes and the allowed diversity. Both methods are analysed experimentally, and quality criteria are proposed in the subsequent section.

Design space quality

We introduce relevant measures to quantify the quality of a subspace with respect to its ability to produce a wide range of diverse and valid shapes. The diversity measure is based on the Hausdorff distance [3], which is widely used to measure how far two subsets of a metric space are from each other. Therefore, it can also be used to measure the similarity/diversity between two free-form shapes. Consider now an instance of $\boldsymbol{v}$ that modifies the parent design $\mathcal{G}$ to $\mathcal{G}'$. Both $\mathcal{G}$ and $\mathcal{G}'$ can be then discretised by an appropriately dense point set, $\mathbf{O} = \{\mathbf{o}_i, i \in \{1, 2, \ldots, n_o\}\} \in \mathcal{G} \subseteq \mathbb{R}^3$ and $\mathbf{O}' = \{\mathbf{o}'_j, j \in \{1, 2, \ldots, n'_o\}\} \in \mathcal{G}' \subseteq \mathbb{R}^3$ containing a total n_o and n'_o points, respectively. The Hausdorff distance, H, between $\mathbf{O}$ and $\mathbf{O}'$ can be then evaluated as

$$H\left(\mathbf{O}, \mathbf{O}'\right) = \max\left\{\sup_{\mathbf{o}\in\mathbf{O}} d(\mathbf{o}, \mathbf{O}), \sup_{\mathbf{o}'\in\mathbf{O}'} d(\mathbf{O}, \mathbf{o}')\right\}, \tag{6.21}$$

where $d(\mathbf{o}, \mathbf{O}') = \inf_{\mathbf{o}'\in\mathbf{O}'} d(\mathbf{o}, \mathbf{o}')$ quantifies the distance from a point $\mathbf{o} \in \mathbf{O}$ to the set $\mathbf{O}'$. Note that both $\mathbf{O}$ and $\mathbf{O}'$ need to be subsets of the same metric space. In our case, we use the Euclidean distance for $d(\mathbf{o}, \mathbf{o}')$ (and $d(\mathbf{o}', \mathbf{o})$), and we further assume that the Hausdorff distance between $\mathcal{G}$ and $\mathcal{G}'$ is quantified by $H\left(\mathbf{O}, \mathbf{O}'\right)$, i.e., $H\left(\mathcal{G}, \mathcal{G}'\right) \coloneqq H\left(\mathbf{O}, \mathbf{O}'\right)$. Hence, we define the diversity measure to be the *average of the Hausdorff distance between the parent design and a dense set of designs sampled from the subspace* $\mathcal{V}$; as illustrated in Fig. 6.5. Therefore, the higher the value of the diversity measure, the richer the subspace. However, as mentioned before, a more diverse design space may also have a high possibility

of producing invalid geometries. A typical example of invalid free-form geometries is that of self-intersecting surfaces. An ideal subspace will have the highest diversity and few invalid geometries. Therefore, we define the validity measure as *the ratio of invalid over valid designs for a dense sampling of* $\mathcal{V}$. Obviously, subspaces with a validity measure equal to or close to 0 are preferred.

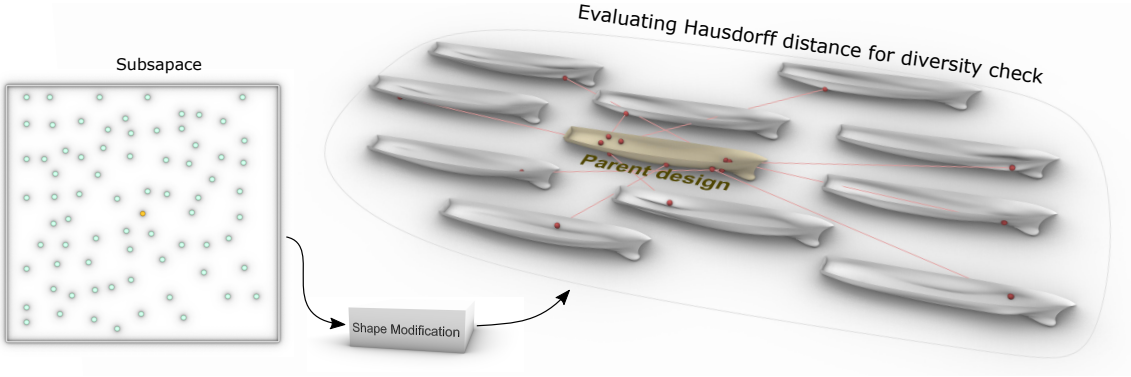


Figure 6.5: Illustration of evaluation of Hausdorff distance for diversity between the parent design and the designs samples from the subspace. The right image's red points indicate where the maximum Hausdorff distance accrues between the parent and sampled design.

Algorithm 3 briefly summarises the step-wise procedure of the proposed approach from dimension reduction to design space formulation and shape optimisation.

6.3 Test Cases

We used two different 3D modelling cases, a wing model based on the NACA 2410 aerofoil[1] (see Fig. 6.6) and a US Navy Combatant DTMB 5415 hull model[2] (see Fig. 5.2 (b)), to analyse and validate the proposed approach. We use the wing model to demonstrate the capability of the proposed approach to generate subspaces with high representation capacity and compactness. The hull model case goes one step further to demonstrate that the proposed approach produces subspaces with high representation capacity and compact-

[1]see, for example, `http://airfoiltools.com/airfoil/details?airfoil=naca2410-il` for more information on NACA 2410 profile.

[2]see, for example, `http://www.simman2008.dk/5415/combatant.html` for more details on DTMB 5415.

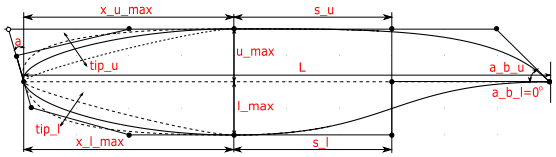
Algorithm 3 Step-wise procedure for implementing the proposed approach.

1: Create an initial model $\mathcal{G}$ composed of coordinate set $\bar{\boldsymbol{\vartheta}} \in \mathcal{G} \subseteq \mathbb{R}^{\bar{n}}$ and parametrise it with n design parameters $(t_1, t_2, \ldots, t_n)$.
2: Define the design space $\mathcal{T}$ with lower and upper bounds of n parameters, $\mathcal{T} := \{t_i^l \leq t_i \leq t_i^u,\ \forall i \in \{1, 2, \ldots n\}\}$.
3: Define a shape modification vector $\boldsymbol{G}(\bar{\boldsymbol{\vartheta}}, \mathbf{t}) \in \mathbb{R}^{n_G}$ to modify $\mathcal{G}$ for any realisation $\mathbf{t} \in \mathcal{T}$.
4: Evaluate geometric moment invariant vector $\boldsymbol{MI}^s \in \mathbb{R}^{n_M}$ of order s containing $n_M = (s+1)(s+2)/2$ components using Eq. (5.2).
5: Defined combined geometry and moment shape signature vector $\boldsymbol{P}(\boldsymbol{\vartheta}, \mathbf{t}) \in \mathbb{R}^{n_P}$, $n_P = n_G + n_M$, in the domain $\mathcal{P} := \mathcal{G} \cup \mathcal{M}$ with $\boldsymbol{\vartheta} = (\bar{\boldsymbol{\vartheta}}, \boldsymbol{\vartheta}_M)$; see Eq. (6.3).
6: Find the mean and variance of SSV using Eq. (6.5) and (6.6), respectively.
7: Employ the KLE to find an optimal linear representation of SSV in Eq. (6.7) while recasting the problem in Eq. (6.11) as an eigenproblem resulting $\mathbf{AW} = \mathbf{W}\boldsymbol{\Lambda}$.
8: Rearrange columns of $\mathbf{W}$, which represents KL-modes/eigenvectors, based on their associated KL-values/eigenvalues, such that $\lambda_i \geq \lambda_{i+1}$.
9: Identify first m KL-modes capturing minimum 95% of the variance based on Eq. (6.13).
10: Form geometrically- and functionally-active latent variable vector $\boldsymbol{v} = \{v_i, i = 1, 2, \ldots, m\}$ as in Eq. (6.18), where $m < n$.
11: With $\boldsymbol{v}$ create a subspace $\mathcal{V} \subset \mathbb{R}^m$ as, $\mathcal{V} := \{\boldsymbol{v} : v_i^l \leq v_i \leq v_i^u,\ \forall i \in \{1, 2, \ldots m\}\}$, where v_i^l and v_i^u are the lower and upper bounds set using either Eq. (6.19) or (6.20).
12: Solve Eq. (6.2) to find an optimal design $\boldsymbol{v}^*$ in $\mathcal{V}$.

ness and is also physics-informed, as the physical QoI is dependent on geometric moments. Therefore, we can significantly expedite the shape optimisation process. The particulars, parameters and solvers used for the DTMB are the same as described in §5.3.2 and 5.3.3, whereas the 3D wing model is based on NACA 2410 aerofoil sections, parameterised via the approach described in [151]; see also Fig. 6.6(a). This parameterisation uses 12 parameters to define a foil profile. The construction of the aerofoil commences with the definition of four simple cubic Bézier curves employed to create the final cubic B-spline curve. The foil's chord length (L) is the only dimensional parameter, and all remaining length parameters are non-dimensionalised by it and vary between $[0, 1]$ while always guaranteeing a valid aerofoil shape instance. Readers are advised to refer to [151] for details on the construction and the parametric definition of the aerofoil. The wing, shown in Fig. 6.6(b), is constructed using three independent aerofoil sections placed at the root, mid-span, and tip of the wing, which follow an appropriate chord-length distribution along the span-wise direction. A fixed sweep angle is used, and the final NURBS surface, representing the wing shape, is generated by a cubic lofting operation. The principal dimensions of the wing, i.e., span length and swept angle, are kept fixed and set to 1.2 meters and 4.29°, respectively. The chord length L at the root and tip is equal to 0.15 and 0.06 meters, respectively. The remaining shape parameters, $n = 3 \times 11$, are defined to reconstruct NACA 2410 profiles for the parent design and are considered free parameters for the design space. Finally, to initiate the DR, the entire surface is discretised with $E = 90 \times 25$ nodes by directly evaluating them on the NURBS surface of the wing.

6.4 Results and discussion

This section discusses the results of extensive experimentation with the proposed approach to analyse its performance and prove its capability for efficient dimensionality reduction compared to other existing methods.



(a)

(b)

Figure 6.6: (a) Parametric representation of the aerofoil defined with 12 different parameters (highlighted in red). (b) 3D wing model constructed with three aerofoils, resulting in $n = 33$ design parameters, is used as a test case for validating the proposed approach.

6.4.1 Evaluation of geometric moment invariants

Geometric moments and their invariants of any order can be calculated for geometries satisfying the conditions indicated in Section 5.2.2. However, high-order geometric moments can be sensitive to noise [118] while, at the same time, numerical inaccuracies are ever-present when evaluating high-order terms [116]. Furthermore, a literature review in various application areas, ranging from kinetic equations [113] to shape retrieval [132], reveals that moments of an order higher than four are rarely helpful. In this connection, we limited the order of geometric moments invariants appearing in SSV up to $s = 4$. The 0th−, 1st, 2nd−, 3rd− and 4th−order geometric moments have n_M equal to 1, 3, 6, 10 and 15 components, respectively. The moment invariants for the wing and the submerged part of the hull are presented in Tables 6.1 and 6.2, respectively. Due to symmetries in shape, any vanishing geometric moment invariants are not added to SSV.

6.4.2 Dimension reduction

The proposed DSDR approach commences with the definition of bounding limits for parameters in $\mathcal{T}$, which for the wing and hull models are assigned to $0 \leq \mathbf{t} \leq 1$ and $-1.02 \leq \mathbf{t} \leq 1.02$, respectively. According to [12, 151], these values provide sufficient variation with a relatively large number of valid shapes. During dimension reduction, the

Table 6.1: Geometric moment invariants up to 4th−order evaluated for the baseline wing shape.

$MI_{0,0,0}$	$MI_{1,0,0}$	$MI_{0,1,0}$	$MI_{0,0,1}$	$MI_{2,0,0}$
1	0	0	0	9.927E-02
$MI_{0,2,0}$	$MI_{0,0,2}$	$MI_{1,1,0}$	$MI_{0,1,1}$	$MI_{1,0,1}$
8.922E-04	10.268	-3.222E-04	-1.213E-02	4.482E-01
$MI_{0,0,3}$	$MI_{0,1,2}$	$MI_{0,2,1}$	$MI_{0,3,0}$	$MI_{1,0,2}$
18.353	-2.170E-02	-1.004E-03	7.692E-06	8.018E-01
$MI_{1,1,1}$	$MI_{1,2,0}$	$MI_{2,0,1}$	$MI_{2,1,0}$	$MI_{3,0,0}$
-1.191E-03	-1.012E-04	-5.843E-02	-4.146E-04	-4.479E-03
$MI_{0,0,4}$	$MI_{0,1,3}$	$MI_{0,2,2}$	$MI_{0,3,1}$	$MI_{0,4,0}$
2.415E+02	-2.855E-01	7.848E-03	-3.353E-05	1.925E-06
$MI_{1,0,3}$	$MI_{1,1,2}$	$MI_{1,2,1}$	$MI_{1,3,0}$	$MI_{2,0,2}$
10.553	-1.070E-02	4.311E-04	-8.069E-07	1.143
$MI_{2,1,1}$	$MI_{2,2,0}$	$MI_{3,0,1}$	$MI_{3,1,0}$	$MI_{4,0,0}$
-4.585E-04	7.540E-05	9.955E-02	-3.682E-05	2.257E-02

Table 6.2: Geometric moment invariants up to 4th−order evaluated for the baseline hull shape.

$MI_{0,0,0}$	$MI_{1,0,0}$	$MI_{0,1,0}$	$MI_{0,0,1}$	$MI_{2,0,0}$
1	0	0	0	2.315
$MI_{0,2,0}$	$MI_{0,0,2}$	$MI_{1,1,0}$	$MI_{0,1,1}$	$MI_{1,0,1}$
4.197E-02	6.984E-03	0	0	-2.378E-02
$MI_{0,0,3}$	$MI_{0,1,2}$	$MI_{0,2,1}$	$MI_{0,3,0}$	$MI_{1,0,2}$
-3.303E-04	0	1.076E-03	0	2.786E-03
$MI_{1,1,1}$	$MI_{1,2,0}$	$MI_{2,0,1}$	$MI_{2,1,0}$	$MI_{3,0,0}$
0	-9.078E-03	2.452E-03	0	4.404E-01
$MI_{0,0,4}$	$MI_{0,1,3}$	$MI_{0,2,2}$	$MI_{0,3,1}$	$MI_{0,4,0}$
1.333E-04	0	2.258E-04	0	3.997E-03
$MI_{1,0,3}$	$MI_{1,1,2}$	$MI_{1,2,1}$	$MI_{1,3,0}$	$MI_{2,0,2}$
-8.841E-04	0	-5.538E-04	0	2.298E-02
$MI_{2,1,1}$	$MI_{2,2,0}$	$MI_{3,0,1}$	$MI_{3,1,0}$	$MI_{4,0,0}$
0	6.045E-02	-2.238E-01	0	12.370

ensemble averages, $\langle\cdot\rangle$ (in Eq. (6.5)), over $\mathcal{T}$ is evaluated using Monte Carlo sampling, with statistically converged number of samples $\Psi = 9000$, $\{\mathbf{t}_\psi\}_{\psi=1}^{\Psi} \sim \rho(\mathbf{t})$. $\rho(\mathbf{t})$ is selected to be a uniform distribution. Thus each shape in $\mathcal{T}$ has the same possibility of being optimal. The lth component of $\{\overline{P}_l(\boldsymbol{\vartheta}, \mathbf{t}_\psi)\}$, namely $\mathrm{d}_l(\mathbf{t}_\psi)$, which is discretised deviation from the mean SSV, is evaluated as

$$\mathrm{d}_l(\mathbf{t}_j) = \{P_l(\boldsymbol{\vartheta}, \mathbf{t}_\psi\} - \frac{1}{\Psi}\sum_{\psi=1}^{\Psi}\{P_l(\boldsymbol{\vartheta}, \mathbf{t}_\psi\}, \tag{6.22}$$

which for all the samples gives a matrix $D_l = \big[\mathrm{d}_l(\mathbf{t}_\psi), \forall j = 1, 2, 3 \ldots \Psi\big]$. Using this, the sub-matrix in Eq. (6.16) can be evaluated as

$$\mathbf{C}_{lk} = \frac{1}{\Psi} D_l D_k^T. \tag{6.23}$$

Similarly, all the components of $\mathbf{C}_{lk},\ l, k = 1, \ldots, n_P$ are evaluated to compute $\mathbf{A}$. Now, in the discrete form, the quality of lower-dimensional representation $\overline{\boldsymbol{P}}(\boldsymbol{\vartheta}, \boldsymbol{v})$ can be assessed via the reconstruction error, measured by the Normalised Mean Squared Error (NMSE) as

$$NMSE = \frac{\sum_{\psi=1}^{\Psi} \left\|\mathbf{d}(\mathbf{t}_\psi) - \mathbf{d}(\boldsymbol{v}_\psi)\right\|^2}{\sum_{\psi=1}^{\Psi} \left\|\mathbf{d}(\mathbf{t}_\psi)\right\|^2}. \tag{6.24}$$

Different subspaces with varying SSVs are constructed to test and analyse the proposed approach's performance. The employed SSVs contain either a single high order vector, i.e., $\boldsymbol{MI}^s \in \mathbb{R}^{n_M}$ with $s = 2/3/4$ and $n_M = 6/10/15$, respectively, and their combinations specifically for the hull model. In other words, the tested SSVs are the following: $\big(\boldsymbol{G}(\boldsymbol{\vartheta}, \mathbf{t}), \boldsymbol{MI}^2\big)$, $\big(\boldsymbol{G}(\boldsymbol{\vartheta}, \mathbf{t}), \boldsymbol{MI}^3\big)$, $\big(\boldsymbol{G}(\boldsymbol{\vartheta}, \mathbf{t}), \boldsymbol{MI}^4\big)$, $\big(\boldsymbol{G}(\boldsymbol{\vartheta}, \mathbf{t}), \boldsymbol{MI}^2, \boldsymbol{MI}^3\big)$, $\big(\boldsymbol{G}(\boldsymbol{\vartheta}, \mathbf{t}), \boldsymbol{MI}^2, \boldsymbol{MI}^4\big)$, $\big(\boldsymbol{G}(\boldsymbol{\vartheta}, \mathbf{t}), \boldsymbol{MI}^3, \boldsymbol{MI}^4\big)$ and $\big(\boldsymbol{G}(\boldsymbol{\vartheta}, \mathbf{t}), \boldsymbol{MI}^2, \boldsymbol{MI}^3, \boldsymbol{MI}^4\big)$, which form the shape-supervised subspaces $\mathcal{V}_{\boldsymbol{G},\boldsymbol{MI}^2}$, $\mathcal{V}_{\boldsymbol{G},\boldsymbol{MI}^3}$, $\mathcal{V}_{\boldsymbol{G},\boldsymbol{MI}^4}$, $\mathcal{V}_{\boldsymbol{G},\boldsymbol{MI}^{2,3}}$, $\mathcal{V}_{\boldsymbol{G},\boldsymbol{MI}^{2,4}}$, $\mathcal{V}_{\boldsymbol{G},\boldsymbol{MI}^{3,4}}$ and $\mathcal{V}_{\boldsymbol{G},\boldsymbol{MI}^{2,3,4}}$, respectively. For the wing model only $\mathcal{V}_{\boldsymbol{G},\boldsymbol{MI}^2}$, $\mathcal{V}_{\boldsymbol{G},\boldsymbol{MI}^3}$ and $\mathcal{V}_{\boldsymbol{G},\boldsymbol{MI}^4}$ are tested. The comparison of these subspaces, in terms of their diversity/richness and validity/robustness, along with their capacity to generate optimal designs, will help us analyse the correlation of each

moment (and moment combinations) with shapes' performance. The performance of these shape-supervised subspaces is also compared with $\mathcal{V}_{\boldsymbol{G}}$ that does not employ any moment-based information. As explained in §5.3.1, the hull's wave-making resistance coefficient, C_w, strongly depends on geometric moments. Therefore, in this case, the performance of the subspaces mentioned above is also compared with $\mathcal{V}_{\boldsymbol{G},C_w}$, which augments geometry, $\boldsymbol{G}(\boldsymbol{\vartheta}, \mathbf{t})$, with the calculated value of C_w.

As previously mentioned, the employed grid for the baseline wing and the hull is composed of $E = 25 \times 90$ nodes, which, along with $n_G = 3$ and the moments, when provided, will produce the matrices, $\mathbf{A}$, in Eq. (6.15). Specifically, the construction of $\mathcal{V}_{\boldsymbol{G}}$, $\mathcal{V}_{\boldsymbol{G},C_w}$, $\mathcal{V}_{\boldsymbol{G},\boldsymbol{MI}^2}$, $\mathcal{V}_{\boldsymbol{G},\boldsymbol{MI}^3}$, $\mathcal{V}_{\boldsymbol{G},\boldsymbol{MI}^4}$, $\mathcal{V}_{\boldsymbol{G},\boldsymbol{MI}^{2,3}}$, $\mathcal{V}_{\boldsymbol{G},\boldsymbol{MI}^{2,4}}$, $\mathcal{V}_{\boldsymbol{G},\boldsymbol{MI}^{3,4}}$ and $\mathcal{V}_{\boldsymbol{G},\boldsymbol{MI}^{2,3,4}}$ is performed on the basis of an $\mathbf{A}$ matrix with 2250×2250, 2251×2251, 2256×2256, 2260×2260, 2265×2265, 2266×2266, 2271×2271, 2275×2275 and 2281×2281 elements[3], respectively. For the hull model, the weighting function $f(\boldsymbol{\vartheta})$ is defined in a way that only counts nodes belonging to the submerged part of the hull and nodes above the waterline assume a zero weight since they do not play any role in the resistance components considered in our problem. On the contrary, for the wing model, $f(\boldsymbol{\vartheta})$ is set to take into account the entire shape during the implementation of the proposed approach. It should also be noted that vector spaces are normalised for both test cases to exhibit the same variance associated with geometry and moment invariants. The selection of active KL-modes (eigenvectors) for the construction of subspaces is performed in a way that guarantees that every subspace retains at least 95% of the variance associated with $\mathcal{T}$. In other words, the number is determined by the sum of KL-values (eigenvalues) that reach this threshold; see Eq. (6.12).

Figures 6.7 and 6.8 depict the percentage of variance retained for the wing and hull models with respect to the dimension of each subspace and the dimension required for each subspace to reach that level. One may easily observe in these figures that all considered augmented subspaces perform much better than the purely geometry-based subspace $\mathcal{V}_{\boldsymbol{G}}$ when assessing variance retention. Successful DSDR requires a subspace retaining

[3]Assuming usage of all moment's components.

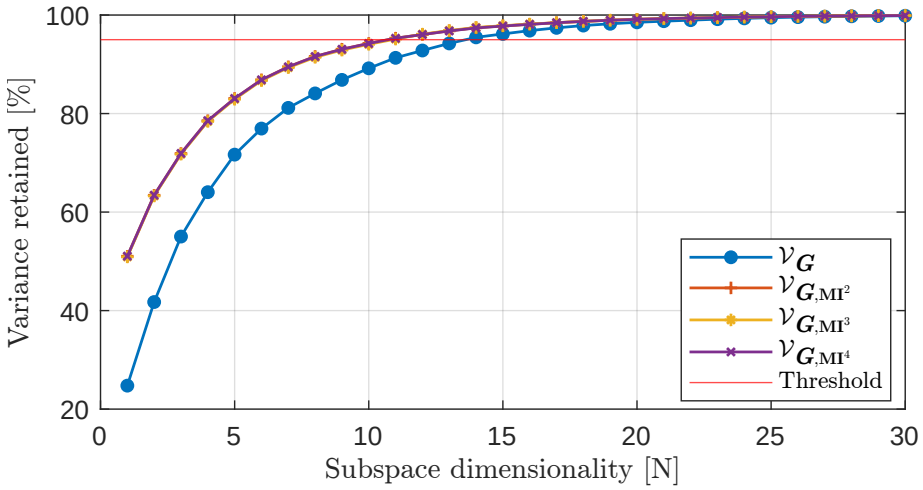


Figure 6.7: Percentage of variance retained by each of the wing model's subspace versus its dimension

. The horizontal red line indicates the 95% threshold.

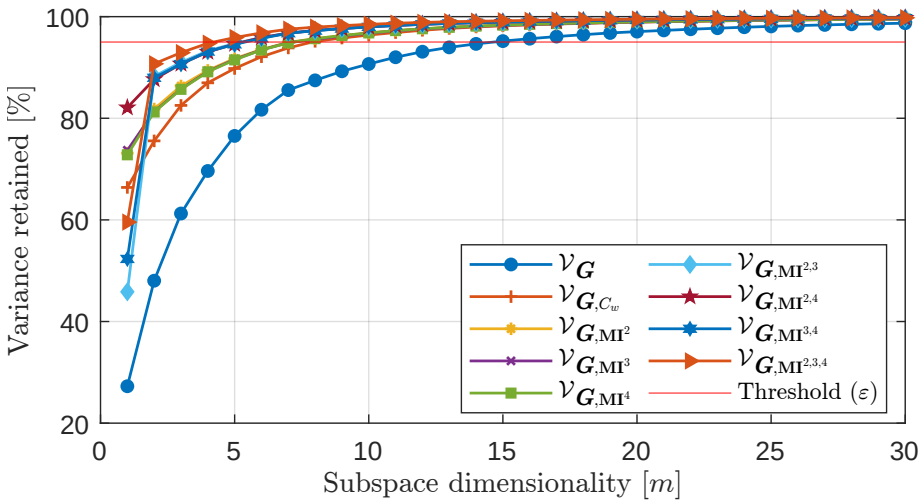


Figure 6.8: Percentage of variance retained by each hull model's subspace versus its dimension. The horizontal red line indicates the 95% threshold.

the highest possible variance with the fewest latent variables. In this aspect, all shape-supervised subspaces reach the threshold variance (95%) with half or fewer dimensions when compared to $\mathcal{V}_{\boldsymbol{G}}$, even if a single component is added to SSV. For the wing model case, shape-supervised subspaces, $\mathcal{V}_{\boldsymbol{G},\boldsymbol{MI}^2}$, $\mathcal{V}_{\boldsymbol{G},\boldsymbol{MI}^3}$ and $\mathcal{V}_{\boldsymbol{G},\boldsymbol{MI}^4}$ have similar performance; requiring $m = 11$ dimensions to capture 95% of variance; thus, resulting in a 67% dimensional reduction, i.e., from $n = 33$ to $m = 11$. More importantly, these subspaces capture higher geometric variance with fewer latent variables compared to solely geometry-based subspace $\mathcal{V}_{\boldsymbol{G}}$, which requires $m = 14$ dimensions for 95% of variance. A more detailed analysis, employing shape-supervised subspaces with moment combinations, is performed for the hull model. In this case, the inclusion of a single geometric moment, i.e., $\boldsymbol{MI}^2$, $\boldsymbol{MI}^3$ or $\boldsymbol{MI}^4$, performs almost identically to C_w's inclusion, which, as stated earlier, confirms the close relation of these moments with C_w. With regards to dimensionality reduction, $\mathcal{V}_{\boldsymbol{G}}$ requires a minimum of $m = 15$ dimensions to capture 95% of variance, which corresponds to a 44% reduction when compared to the original space, $\mathcal{T} \subseteq \mathbb{R}^{27}(n = 27)$. On the other hand, $\mathcal{V}_{\boldsymbol{G},C_w}$, $\mathcal{V}_{\boldsymbol{G},\boldsymbol{MI}^2}$, $\mathcal{V}_{\boldsymbol{G},\boldsymbol{MI}^3}$ and $\mathcal{V}_{\boldsymbol{G},\boldsymbol{MI}^4}$ need $m = 8$, which corresponds to a reduction of 70%, while the spaces using moment combinations exhibit the best performance, i.e., $\mathcal{V}_{\boldsymbol{G},\boldsymbol{MI}^{2,3}}$, $\mathcal{V}_{\boldsymbol{G},\boldsymbol{MI}^{2,4}}$, and $\mathcal{V}_{\boldsymbol{G},\boldsymbol{MI}^{3,4}}$ require $m = 6$ parameters achieving a reduction of 78% and finally $\mathcal{V}_{\boldsymbol{G},\boldsymbol{MI}^{2,3,4}}$ needs only $m = 5$, resulting in the reduction of 81%. This demonstrates the effectiveness of the approach in significantly reducing dimensionality. Finally, we need to note that $\mathcal{V}_{\boldsymbol{G},\boldsymbol{MI}^2}$, $\mathcal{V}_{\boldsymbol{G},\boldsymbol{MI}^3}$ and $\mathcal{V}_{\boldsymbol{G},\boldsymbol{MI}^4}$ achieve the same reduction as $\mathcal{V}_{\boldsymbol{G},C_w}$ which is created in a physics-supervised setting with the inclusion of C_w [33]. Moreover, the construction of $\mathcal{V}_{\boldsymbol{G},C_w}$ is time-consuming as C_w evaluation is computationally expensive, whereas geometric moments have minimal cost. This supports our claim that geometric moments are adequate in capturing the physics involved in our problem, and costly computational approaches can be avoided.

Figures 6.9 and 6.10 show the first three KL-modes, $\mathbf{w}^1$, $\mathbf{w}^2$ and $\mathbf{w}^3$ for all employed subspaces projected on the wing and hull grids, respectively. This projection is of great practical value as it highlights the type and order of variance corresponding to each KL-

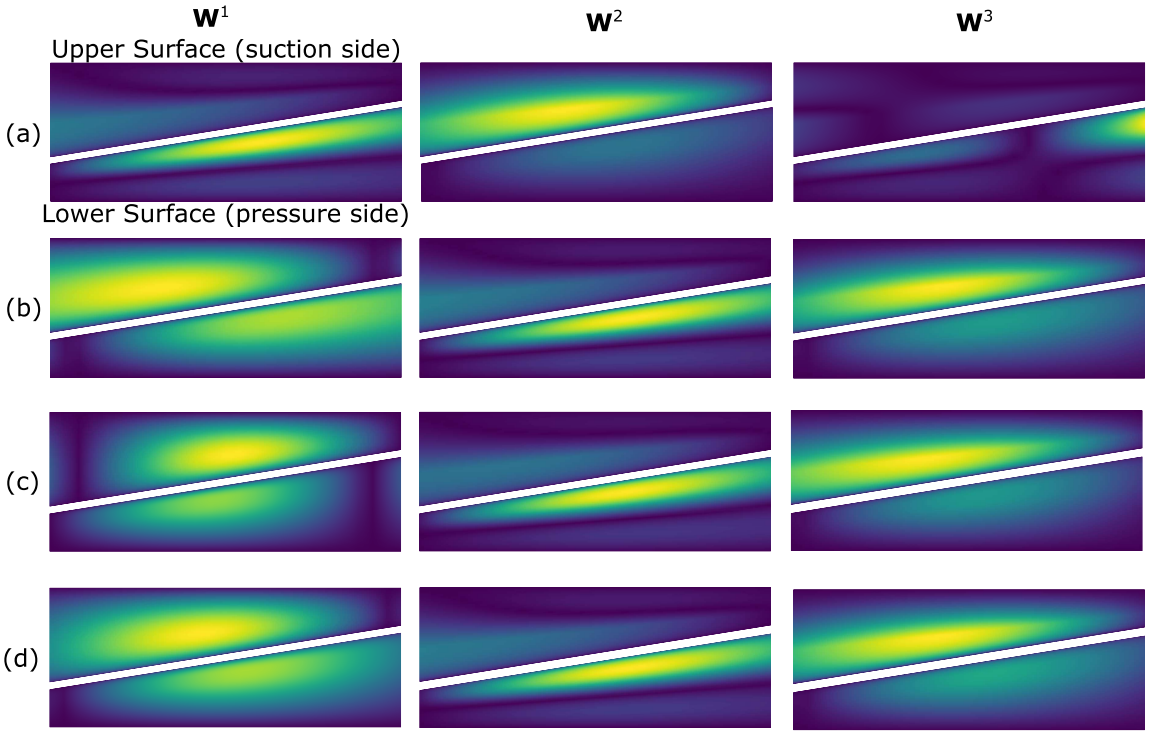


Figure 6.9: Shape deformation of the wing model corresponding to the first three eigenvectors of all employed subspaces: (a) $\mathcal{V}_{\boldsymbol{G}}$ (b) $\mathcal{V}_{\boldsymbol{G},\boldsymbol{MI}^2}$, (c) $\mathcal{V}_{\boldsymbol{G},\boldsymbol{MI}^3}$ and (d) $\mathcal{V}_{\boldsymbol{G},\boldsymbol{MI}^4}$. The magnitude of surface displacement is colour coded [small:blue to large:yellow].

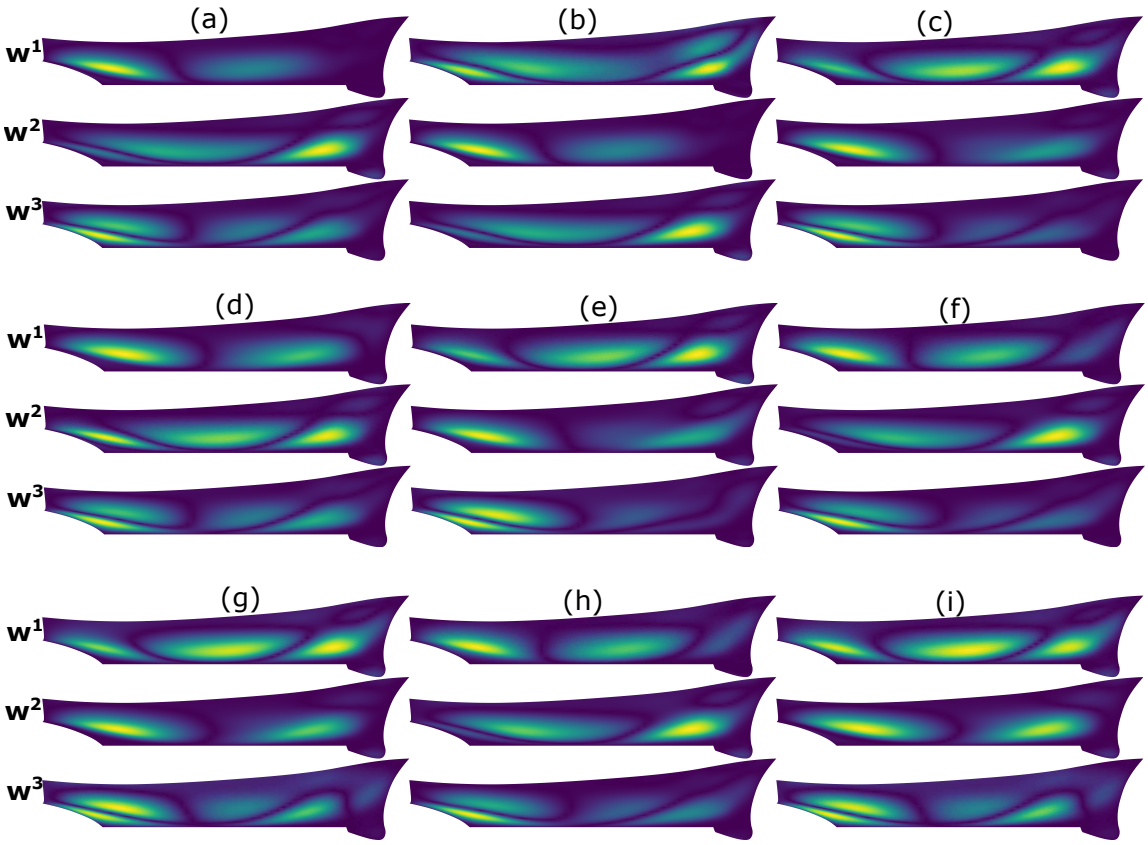


Figure 6.10: Shape deformation of hull model corresponding to the three first eigenvectors of all employed subspaces: (a) $\mathcal{V}_{\boldsymbol{G}}$ (b) $\mathcal{V}_{\boldsymbol{G},C_w}$, (c) $\mathcal{V}_{\boldsymbol{G},\boldsymbol{MI}^2}$, (d) $\mathcal{V}_{\boldsymbol{G},\boldsymbol{MI}^3}$, (e) $\mathcal{V}_{\boldsymbol{G},\boldsymbol{MI}^4}$, (f) $\mathcal{V}_{\boldsymbol{G},\boldsymbol{MI}^{2,3}}$, (g) $\mathcal{V}_{\boldsymbol{G},\boldsymbol{MI}^{2,4}}$, (h) $\mathcal{V}_{\boldsymbol{G},\boldsymbol{MI}^{3,4}}$ and (i) $\mathcal{V}_{\boldsymbol{G},\boldsymbol{MI}^{2,3,4}}$. The magnitude of surface displacement is colour coded [small:blue to large:yellow].

mode. There are some interesting remarks drawn from these figures. From Fig. 6.9, it can be seen that the first ($\mathbf{w}^1$) and third ($\mathbf{w}^3$) KL-modes of $\mathcal{V}_{\boldsymbol{G}}$ show high deviation at the lower surface (pressure side) of the wing, with ($\mathbf{w}^1$) affecting the leading edge and ($\mathbf{w}^3$) affecting the leading edge area near the root. On the contrary, $\mathbf{w}^2$ affects mainly the upper surface. The sets of KL-modes of $\mathcal{V}_{\boldsymbol{G},\boldsymbol{MI}^2}$, $\mathcal{V}_{\boldsymbol{G},\boldsymbol{MI}^3}$ and $\mathcal{V}_{\boldsymbol{G},\boldsymbol{MI}^4}$, are very similar to each other. Specifically, $\mathbf{w}^1$, in all cases, relates to both upper and lower surfaces, whereas, interestingly, $\mathbf{w}^2$ and $\mathbf{w}^3$ are similar to the $\mathbf{w}^1$ and $\mathbf{w}^2$ of $\mathcal{V}_{\boldsymbol{G}}$. For the hull model, apart from Fig. 6.10(b), variation is exhibited only below the waterline as the proposed method assigns zero weight to nodes above the waterline. The first KL-mode ($\mathbf{w}^1$) of $\mathcal{V}_{\boldsymbol{G},C_w}$ is highly affected by the inclusion of physics, i.e., C_w, whereas the remaining two ($\mathbf{w}^2$ and $\mathbf{w}^3$) are identical to $\mathbf{w}^1$ and $\mathbf{w}^2$ of $\mathcal{V}_{\boldsymbol{G}}$. This pattern persists for the remaining higher modes, not depicted in the figure. In the case of shape-supervised subspaces for the hull model, results can be grouped in two sets, $\{\mathcal{V}_{\boldsymbol{G},\boldsymbol{MI}^2}, \mathcal{V}_{\boldsymbol{G},\boldsymbol{MI}^4}, \mathcal{V}_{\boldsymbol{G},\boldsymbol{MI}^{2,4}}\}$ and $\{\mathcal{V}_{\boldsymbol{G},\boldsymbol{MI}^3}, \mathcal{V}_{\boldsymbol{G},\boldsymbol{MI}^{2,3}}, \mathcal{V}_{\boldsymbol{G},\boldsymbol{MI}^{3,4}}, \mathcal{V}_{\boldsymbol{G},\boldsymbol{MI}^{2,3,4}}\}$ as their respective KL-modes bare noticeable similarities. Although both sets exhibit different first and second modes when compared with $\mathcal{V}_{\boldsymbol{G}}$, the third mode, i.e., $\mathbf{w}^3$, is very similar along all cases but $\mathcal{V}_{\boldsymbol{G},\boldsymbol{C_w}}$ which pushed down that mode to become $\mathbf{w}^4$. However, $\mathbf{w}^1$ of the second set bears some resemblance to $\mathbf{w}^1$ of $\mathcal{V}_{\boldsymbol{G}}$ while the first set seems to more closely follow the $\mathbf{w}^2$ of $\mathcal{V}_{\boldsymbol{G}}$. Fig. 6.12 and 6.11 depict NMSE (see Eq. (6.24)) versus subspace dimensionality for the wing and hull models, respectively. NMSE reduces for all subspaces as their dimension m increases. Except for $\mathcal{V}_{\boldsymbol{G},C_w}$ in the case of the hull model, there is no significant difference between the NMSE of the initial subspace $\mathcal{V}(\boldsymbol{G})$ and the remaining shape-supervised subspaces.

6.4.3 Shape-supervised DSDR with composite-SSV for the hull model

For feature-rich and complex geometries like the hull model, geometric moments of higher order (above four) may be needed to capture local features that do affect wave-making resistance if they reside, for example, in the bulbous bow. However, as mentioned earlier, incorporating higher-order moments comes with problems related to noise and numerical

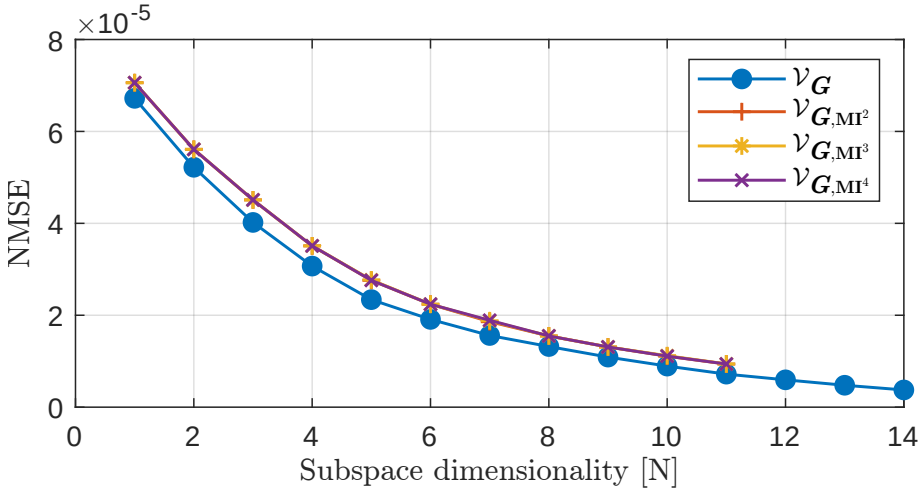


Figure 6.11: Reconstruction accuracy of wing model's subspaces measured via NMSE with respect to their dimensionality (m).

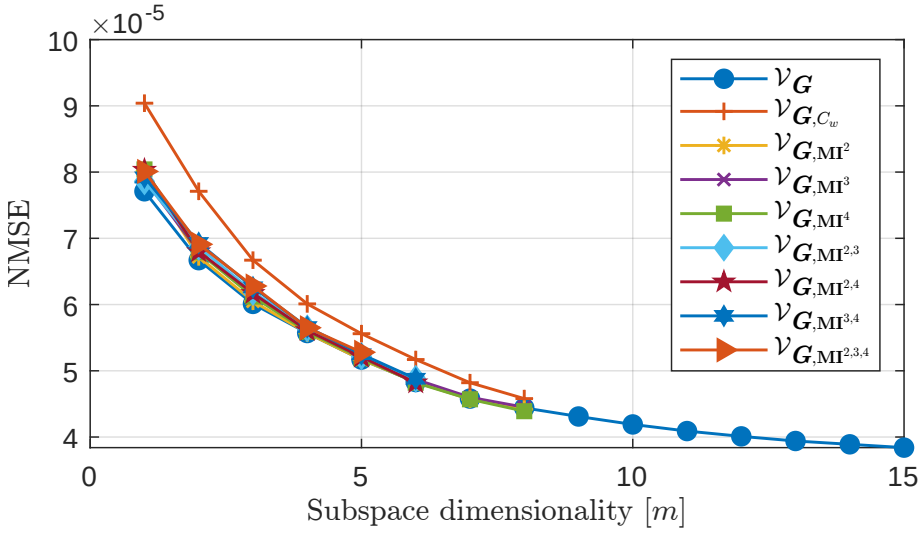


Figure 6.12: Reconstruction accuracy of hull model's subspaces measured via NMSE with respect to their dimensionality (m).

issues. Therefore, instead of evaluating higher-order geometric moments, one may decompose the geometry into sufficiently simple parts so that lower-order moments can efficiently describe them. This decomposition results in a composite moment vector containing up to sth$-$order moment invariants for each part. The corresponding SSV will incorporate the moment composite vector in such cases. Henceforth, in this connection, we shall refer to composite-SSV versus the global-SSV used in the previous section.

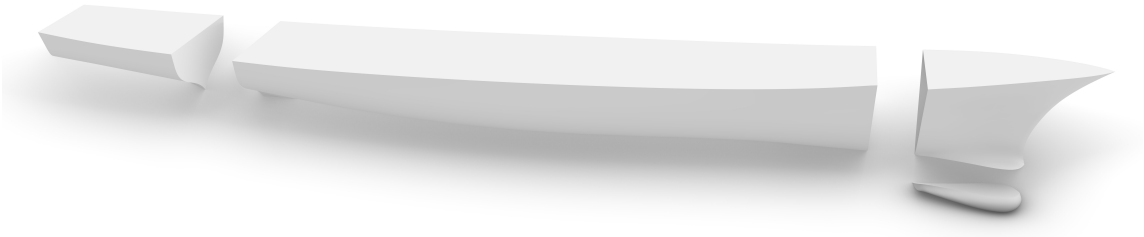


Figure 6.13: Decomposition of hull model for DSDR with composite-SSV.

The hull decomposition we have used is shown in Fig. 6.13. We split the hull model into four parts: sonar dome, for-part, mid-body, and aft-part. After the segmentation, composite-SSV is composed of all sth order moment invariants, $\boldsymbol{MI}^s \in \mathbb{R}^{4n_M}$, $n_M = (s+1)(s+2)/2$, obtained for each of the four segments and the shape modification vector function (G), which is evaluated for the entire shape to ensure smooth deformation over the segments. KLE is then similarly performed on the composite-SSV to global $\boldsymbol{MI}^s$ to obtain a single subspace used for shape optimisation. Fig. 6.14 corresponds to the previously discussed Fig. 6.8. We should also note here that the segmented shape is only used with the shape-supervised subspaces, and therefore the results for $\mathcal{V}_{\boldsymbol{G}}$ and $\mathcal{V}_{\boldsymbol{G},C_w}$ remain unchanged.

Fig. 6.14 depicts a similar pattern, in terms of variance, of the shape-supervised subspaces constructed with composite-SSV to the ones constructed previously with the global-SSV. However, in this case, the variance retained by the first few latent variables is comparably less. For example, at $m = 1$, $\mathcal{V}_{\boldsymbol{G},\boldsymbol{MI}^{2,4}}$ exhibits a variance of approximately 63%, whereas the same space recorded the largest variance (around 82%) in the global case. In

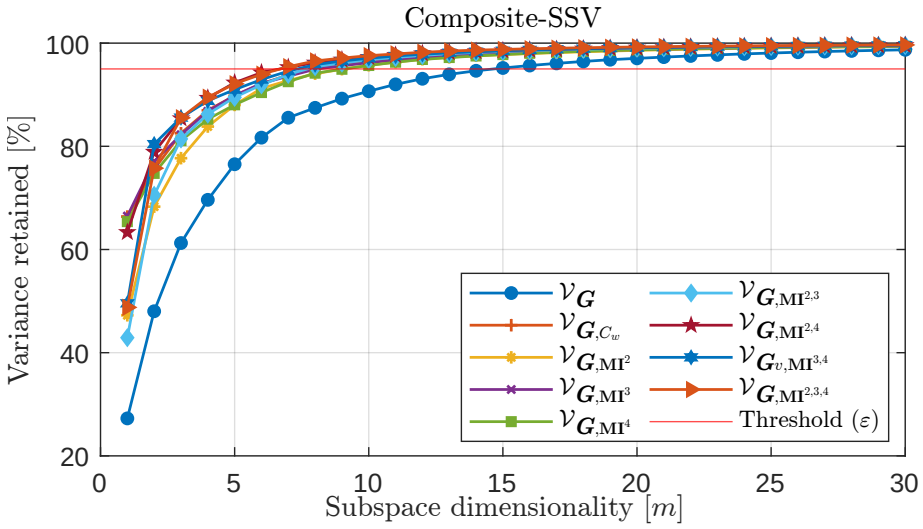


Figure 6.14: Percentage of variance retained by each hull model's subspace versus its dimension. The horizontal red line indicates the 95% threshold.

the composite case, the largest variance is retained by $\mathcal{V}_{\boldsymbol{G},\boldsymbol{MI}^3}$, followed $\mathcal{V}_{\boldsymbol{G},\boldsymbol{MI}^4}$, $\mathcal{V}_{\boldsymbol{G},\boldsymbol{MI}^{2,4}}$ at $m = 1$. It is also interesting to note that the variance retained by $\mathcal{V}_{\boldsymbol{G},C_w}$, overall plotted dimensions, closely matches the variance retained by $\mathcal{V}_{\boldsymbol{G},\boldsymbol{MI}^3}$ in this case. This again demonstrates how moments, especially composite $\boldsymbol{MI}^3$, can capture the behaviour of C_w in the proposed approach for dimensionality reduction.

Fig. 6.15 shows the final dimensionality of all subspaces. In the case of composite-SSV, the dimensionality of the shape-supervised subspaces is higher than what was achieved with global-SSVs. The dimensions of $\mathcal{V}_{\boldsymbol{G},\boldsymbol{MI}^2}$, $\mathcal{V}_{\boldsymbol{G},\boldsymbol{MI}^3}$ and $\mathcal{V}_{\boldsymbol{G},\boldsymbol{MI}^4}$ increased from 8 to 10 (and 9 $\mathcal{V}_{\boldsymbol{G},\boldsymbol{MI}^3}$) and now exhibit a dimensionality reduction of approximately 63%, 67% and 63%, respectively. A significant increase is observed in the case of $\mathcal{V}_{\boldsymbol{G},\boldsymbol{MI}^{2,3,4}}$, whose dimensionality increased from 5 to 7, which now matches the dimensionality of $\mathcal{V}_{\boldsymbol{G},\boldsymbol{MI}^{2,4}}$. Finally, NMSE values for the composite case resemble the results presented in Fig. 6.12; therefore, no separate figure is included here.

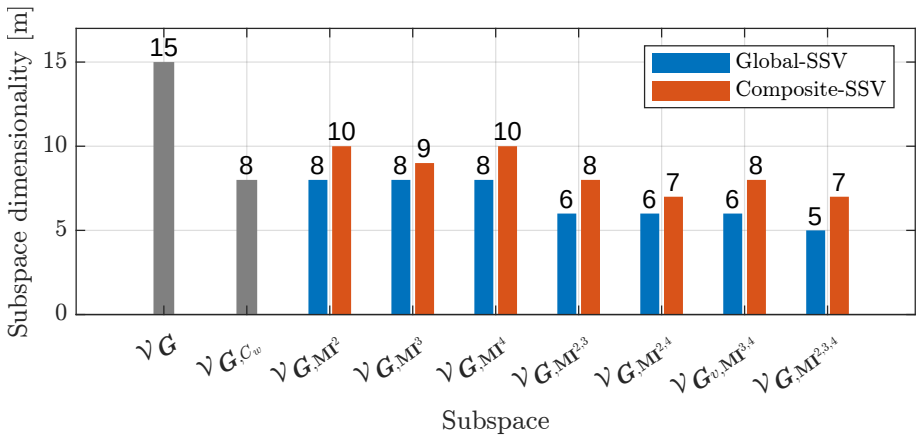


Figure 6.15: Dimension required by each hull model's subspace to reach 95% of the variance threshold.

6.4.4 Subspace quality analysis (SQA)

Once the new basis and the corresponding subspaces are formed for both test cases, their quality for representation capacity and compactness against the criteria defined in §6.2.4 is analysed. The analysis assesses the suitability of the subspace for shape optimisation, i.e., we assess whether the subspace $\mathcal{V}$ resulting from new parametrisation of shapes with latent variables $\boldsymbol{v}$ can capture the underlying shape structure adequately and whether it produces valid and diverse geometries. To commence these analyses, we use five random Monte Carlo samplings of $\Psi = 5,000,000$ parameter vectors from each subspace and compute the average number of invalid shapes (i.e., shapes with self-intersecting geometries) appearing in each subspace. We first briefly analyse the quality of shape-supervised subspaces, $\mathcal{V}_{\boldsymbol{G},\boldsymbol{MI}^2}$, $\mathcal{V}_{\boldsymbol{G},\boldsymbol{MI}^3}$ and $\mathcal{V}_{\boldsymbol{G},\boldsymbol{MI}^4}$ constructed for the wing model and compare them with $\mathcal{V}_{\boldsymbol{G}}$. Afterwards, we perform a detailed Subspace Quality Analysis (SQA) for the hull model's subspaces constructed with global- and composite-SSVs.

SQA for the wing model

Figure 6.16 shows the average number of invalid wing designs and the average diversity of designs present in subspaces $\mathcal{V}_{\boldsymbol{G}}$, $\mathcal{V}_{\boldsymbol{G},\boldsymbol{MI}^2}$, $\mathcal{V}_{\boldsymbol{G},\boldsymbol{MI}^3}$ and $\mathcal{V}_{\boldsymbol{G},\boldsymbol{MI}^4}$ bounded with Eq. (6.19).

From Fig. 6.16(a), it can be seen that the $\mathcal{V}_{\boldsymbol{G}}$ subspace, constructed using only geometry, as in [22], produces a significantly larger number of invalid shapes when compared to the proposed shape-supervised subspaces, i.e., $\mathcal{V}_{\boldsymbol{G},\boldsymbol{MI}^2}$, $\mathcal{V}_{\boldsymbol{G},\boldsymbol{MI}^3}$ and $\mathcal{V}_{\boldsymbol{G},\boldsymbol{MI}^4}$. The average diversity measure, calculated using Eq. (6.21), for the wing case and all subspaces is shown in Fig. 6.16(b). Note that the diversity of designs in $\mathcal{V}_{\boldsymbol{G}}$ is only slightly higher to the ones in $\mathcal{V}_{\boldsymbol{G},\boldsymbol{MI}^2}$, $\mathcal{V}_{\boldsymbol{G},\boldsymbol{MI}^3}$ and $\mathcal{V}_{\boldsymbol{G},\boldsymbol{MI}^4}$, which is practically negligible. These results show that even if there is no prior information on physics or its dependence on geometric moments, the shape-supervised subspaces are significantly more robust in terms of providing valid shapes while maintaining similar levels of design diversity. These capabilities of subspaces are beneficial for accelerating the convergence of shape optimisation towards optimal solution [13].

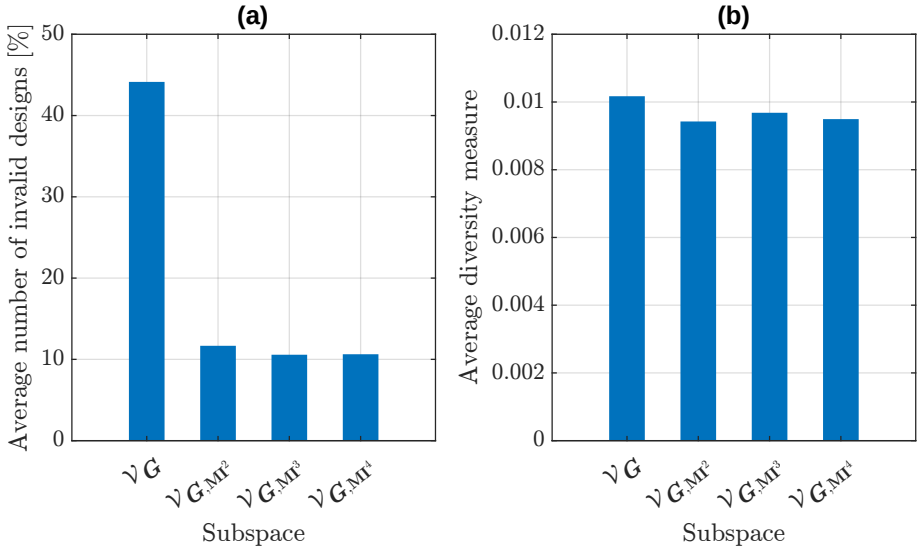


Figure 6.16: (a) Average percent of invalid wing designs and (b) average diversity measure for wing designs in subspaces $\mathcal{V}_{\boldsymbol{G}}$, $\mathcal{V}_{\boldsymbol{G},\boldsymbol{MI}^2}$, $\mathcal{V}_{\boldsymbol{G},\boldsymbol{MI}^3}$ and $\mathcal{V}_{\boldsymbol{G},\boldsymbol{MI}^4}$.

SQA for the hull model

For the hull model, apart from comparing subspaces formed with global- and composite-SSVs, we also assess the effect of the approach employed in setting parameter bounds

(see Eqs. (6.19) and (6.20)) on their quality. We henceforth denote with SL_1 and SL_2 the results following the approach in Eq. (6.19) and (6.20), respectively. The resulting percentages of invalid geometries using SL_1 and SL_2 are shown in Fig. 6.17.

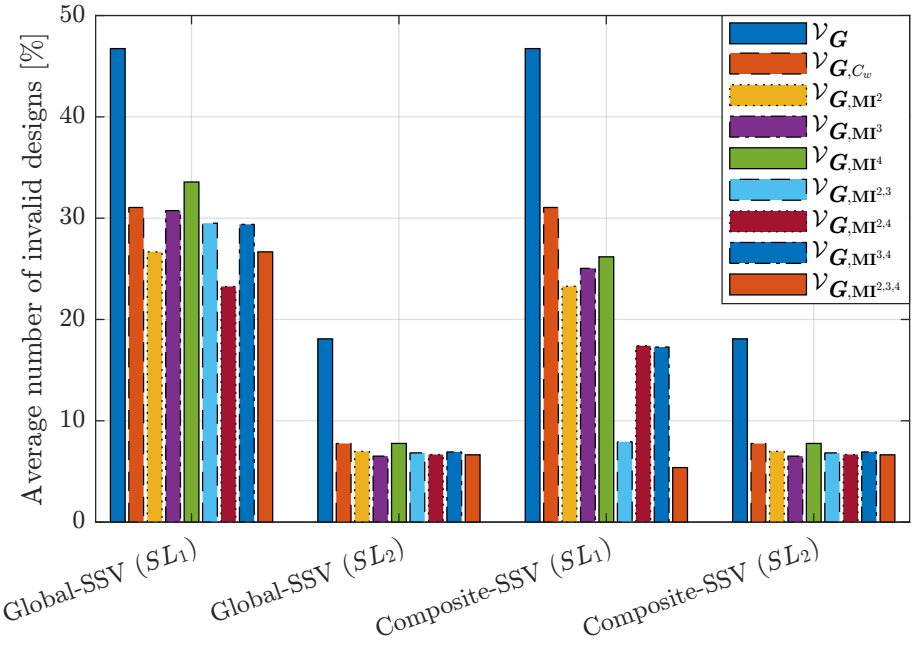


Figure 6.17: Average percent of invalid hull designs in $\mathcal{V}_{\boldsymbol{G}}$, $\mathcal{V}_{\boldsymbol{G},C_w}$ and shape-supervised subspaces sampling with global- and composite-SSVs when bounded by SL_1 and SL_2 approaches.

The following remarks can be drawn by observing the results in Fig. 6.17: i) SL_1 leads to more invalid geometries for all subspaces; ii) shape-supervised subspaces with composite-SSV have a lower percentage of invalid geometries to global-SSV, even when SL_1 is used; iii) the number of invalid geometries in $\mathcal{V}_G$ are substantially higher than any other subspace regardless of the bounding approach; finally iv) in all cases, shape-supervised subspaces tend to produce a similar or even lower number of invalid shapes when compared to $\mathcal{V}_{\boldsymbol{G},C_w}$. These results confirm the ability of shape-supervised subspaces to generate a large number of valid geometries, thereby promoting fast convergence in optimisation and, more importantly, manifesting the ability of geometric moments to attain the performance

of the physics-informed DSDR with C_w without the computational penalty induced by it.

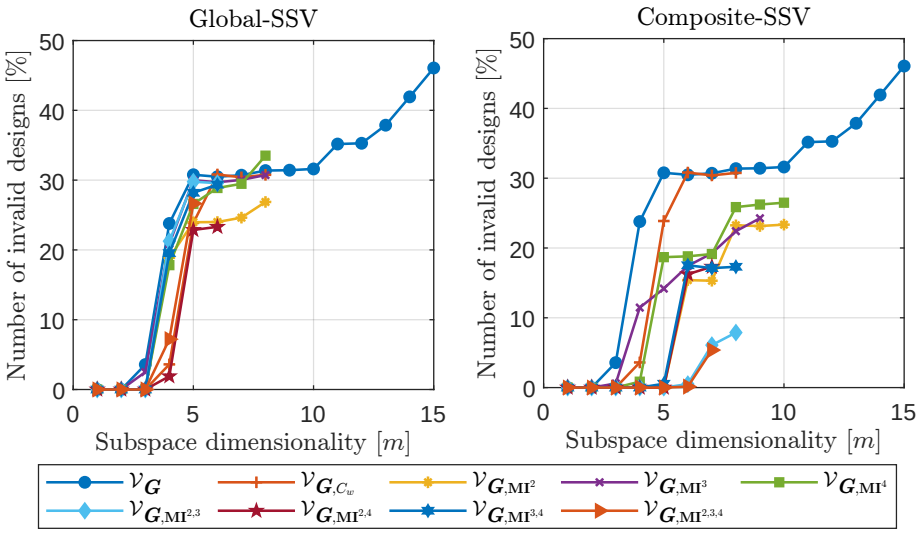


Figure 6.18: Percentage of invalid hull designs as a function of dimensionality of subspace formed with SL_1 bounding approach.

Fig. 6.18 depicts the relation of invalid shapes percentage to dimensionality m when subspaces are formed with SL_1. The first obvious observation is that there are either no or only a few invalid geometries for the first few dimensions, but these increase rapidly after the 4th or 5th dimension. This trend is more prominent for $\mathcal{V}_{\boldsymbol{G}}$: while $m = 2$ shows no invalid geometries, $m = 3$ records an increase to 3.5%, and at $m = 4$ this abruptly goes up to 23% and stabilises to around 30% till $m = 10$ before increasing further. A downward shift can be observed for shape-supervised subspaces with composite-SSV, but, in this case, the relation with dimensionality is also affected. The selection of the SL_2 bounding approach does not affect this relationship.

Finally, we also analysed the diversity of subspaces as described in §6.2.4. The results of the analysis are collectively presented in Fig. 6.19. Similar to the validity analysis, these results are obtained by averaging over 5 Monte-Carlo samplings with a size of $\Psi = 5,000,000$ and diversity is only measured for valid geometries. Most subspaces bounded by SL_1 have a higher diversity index than when bounded by SL_2. Note that despite their

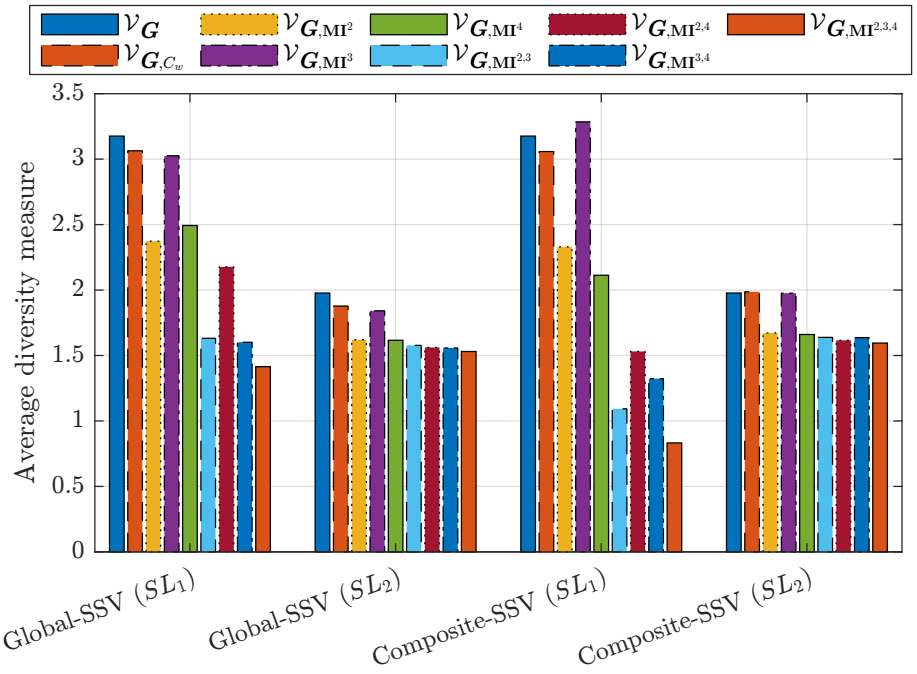


Figure 6.19: Average diversity measure for hull designs in $\mathcal{V}_{\boldsymbol{G}}$, $\mathcal{V}_{\boldsymbol{G},C_w}$ and shape-supervised subspaces created with global- and composite-SSV bounded by SL_1 and SL_2 techniques.

lower dimensionality, shape-supervised subspaces have similar diversity to $\mathcal{V}_{\boldsymbol{G}}$, which is especially true when subspaces, formed with either global- or composite-SSV, are bounded by SL_2. More importantly, $\mathcal{V}_{\boldsymbol{G}}$, $\mathcal{V}_{\boldsymbol{G},C_w}$ and $\mathcal{V}_{\boldsymbol{G},\boldsymbol{MI}^3}$ have similar diversity performance although $\mathcal{V}_{\boldsymbol{G},C_w}$ and $\mathcal{V}_{\boldsymbol{G},\boldsymbol{MI}^3}$ have lower dimensionality and less than half of the invalid shapes when compared to $\mathcal{V}_{\boldsymbol{G}}$. Fig. 6.20 draws a more detailed picture of these results as it depicts diversity performance buildup with subspace dimensionality. Diversity increases monotonically with dimensionality; however, it slowly tends to its maximum value after including a sufficient number of dimensions for each subspace. This observation is in line with our previous analysis in which we indicated that the first few KL-modes (5 to 8) forming the basis of these subspaces capture most of the variance.

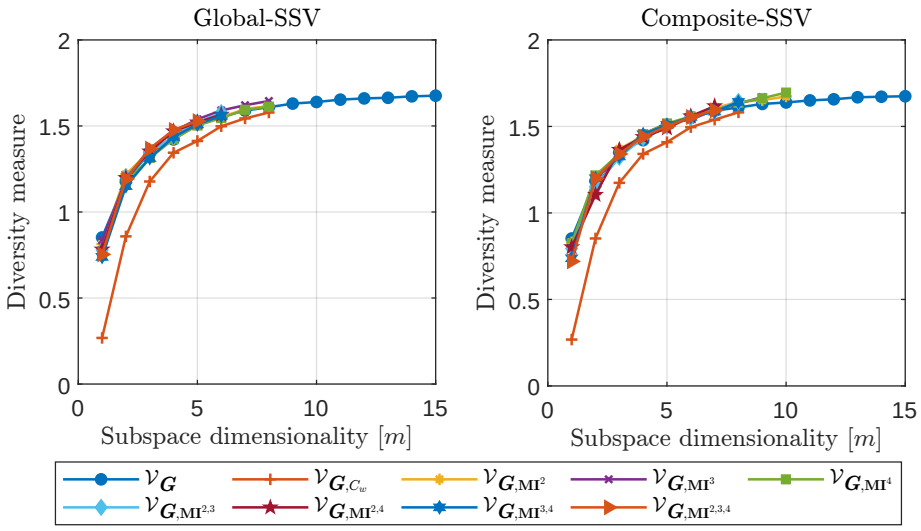


Figure 6.20: Plot showing the diversity measure of hull designs as a function of dimensionality of subspace formed with SL_2 bounding approach.

6.4.5 Shape optimisation of the hull model

When comparing SL_1 with SL_2, we can see that the number of invalid geometries generated by SL_1 is twice as large as SL_2; however, there is no significant difference between the two approaches in terms of diversity. We, therefore, employ SL_2 to set the bounding limits

of the subspace used for shape optimisation. Finally, after performing the dimensionality reduction, the optimisation problem is redefined as follows:

$$\begin{aligned} &\text{Find } \boldsymbol{v}^* \in \mathbb{R}^m \quad \text{such that} \\ &\qquad C_w(\boldsymbol{v}^*) = \min_{\boldsymbol{v}\in\mathcal{V}} C_w(\boldsymbol{v}) \\ &\text{subject to} \quad 0.95V_0 \leq V(\boldsymbol{v}) \leq 1.05V_0, \\ &\qquad\qquad\quad 0.95B_{WL_0} \leq B_{WL}(\boldsymbol{v}) \leq 1.05B_{WL_0}, \\ &\qquad\qquad\quad L_{WL}(\boldsymbol{v}) = L_{WL_0} \text{ and } T(\boldsymbol{v}) = T_0, \end{aligned} \tag{6.25}$$

where V, B_{WL}, L_{WL}, T correspond to volume, length and beam at the waterline, and draft, respectively. The sub-index $(_0)$ indicates the quantity values for the parent hull. The optimisation problem above is solved using Jaya Algorithm (JA), a simple yet efficient optimiser; see more details in [109]. Furthermore, as JA employs a stochastic approach, results may differ in each run; therefore, three different optimisation runs are performed, and the results are averaged in this work. In each run, 150 iterations are considered, and Fig. 6.21 displays the convergence graph over the first 50 iterations. Optimum designs obtained for each case are depicted in Fig. 6.22(a). The contours shown in Fig. 6.22(a) constitute the so-called hulls' *body-plan*. It consists of the halves of cross-sections resulting from intersecting the hull with planes located perpendicularly to its longitudinal symmetry plane. Cross-sections from amidships to the forward part of the hull are drawn on the righthand side of the figure, while the remaining sections, amidships to the stern, are drawn on the left-hand side. An example of the construction of such cross-sections is shown in Fig. 6.22 (b). Cross-sections highlighted in blue correspond to optimised designs, while the ones highlighted in grey are the baseline design. Plotting baselines and optimised designs' cross-sections on the same image facilitates the comparison of their geometrical features. Such a comparison is widely used in naval architecture. The QoI value, i.e., the wave resistance of the hull, is the criterion for deciding which of the two is the best hull.

The convergence graph in Fig. 6.21 demonstrates the competitive performance of the

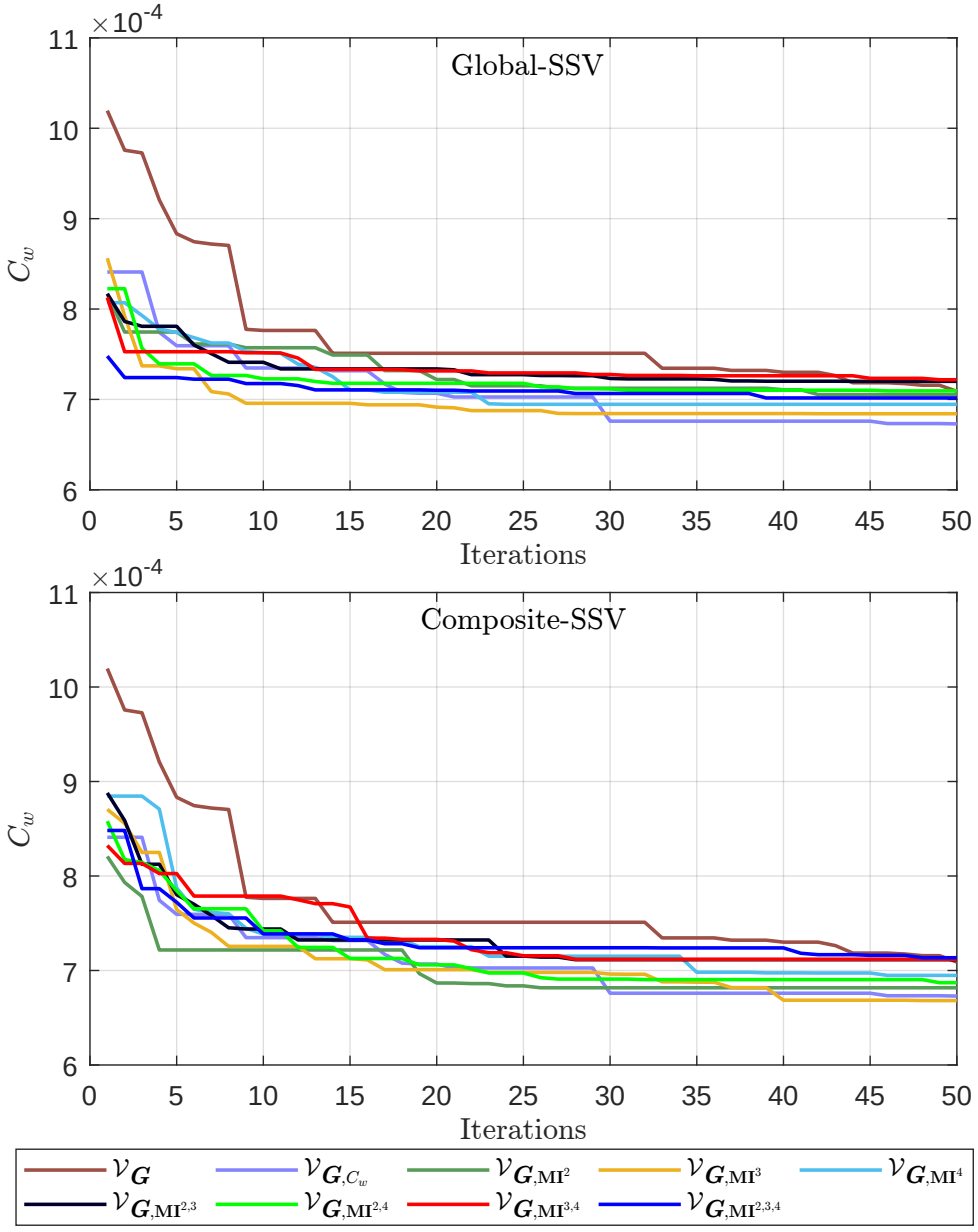


Figure 6.21: C_w optimisation history for $\mathcal{V}_{\boldsymbol{G}}$, $\mathcal{V}_{\boldsymbol{G},C_w}$, and the shape-supervised subspaces with global- and composite-SSV.

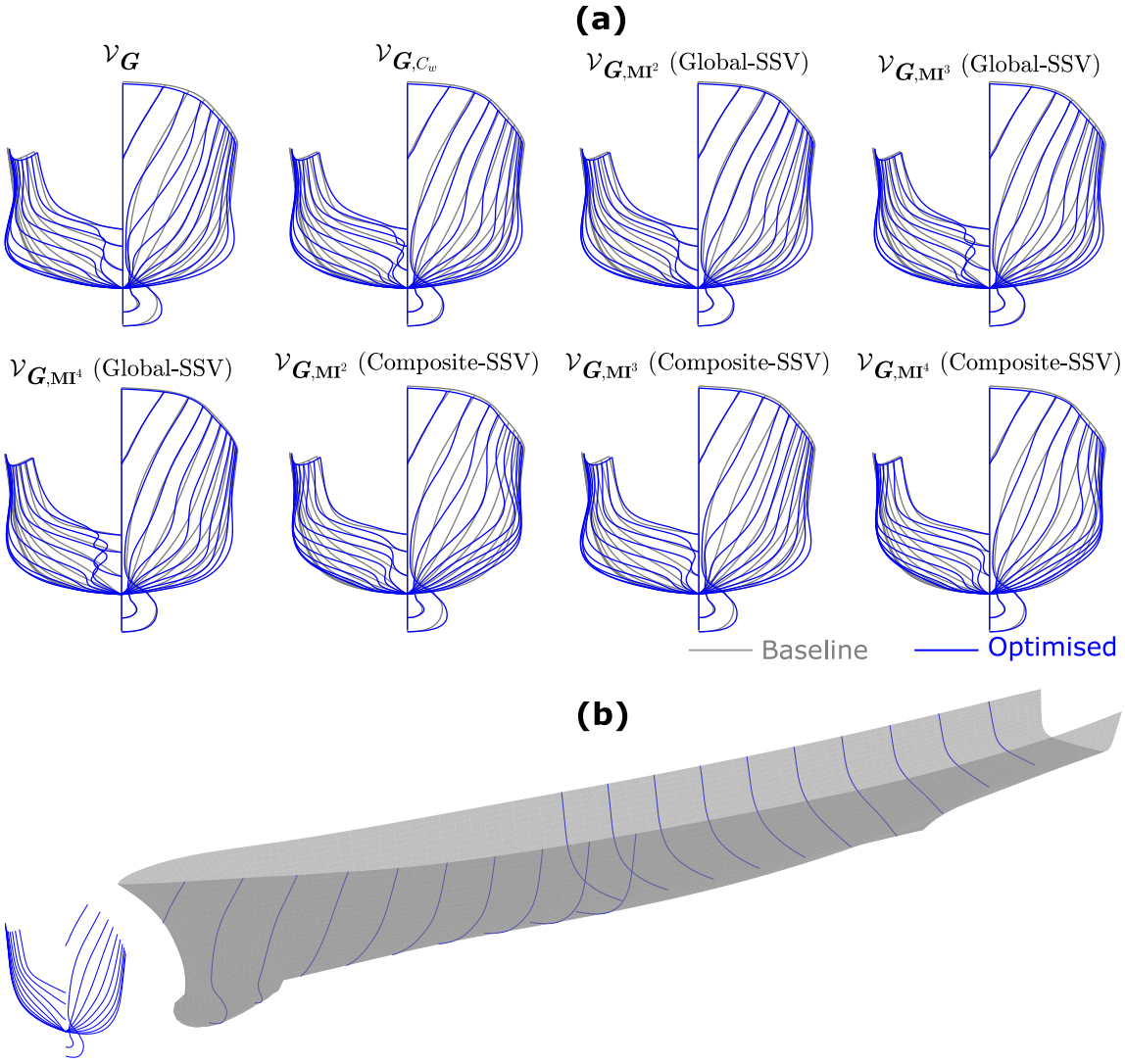


Figure 6.22: (a) Comparison between the baseline and optimised hull shapes, in terms of cross-sections (or body-plan), obtained at the end of the optimisation process. (b) Example of construction of hull's cross-sections.

proposed approach. Shape optimisation performed with shape-supervised subspaces (both global- and composite-SSV) converge substantially faster than $\mathcal{V}_{\boldsymbol{G}}$ and, more importantly, exhibits a similar convergence performance with $\mathcal{V}_{\boldsymbol{G},C_w}$. One of the reasons for the slower convergence of $\mathcal{V}_{\boldsymbol{G}}$ is the existence of many invalid shapes, whereas the remaining subspaces perform better from the very beginning. All shape-supervised spaces tend to approach the performance of $\mathcal{V}_{\boldsymbol{G},C_w}$ and especially, $\mathcal{V}_{\boldsymbol{G},\boldsymbol{MI}^3}$ that even surpasses it when built with composite-SSV.

Table 6.3: Average C_w values over three optimisation runs after 150 iterations.

Design Spaces	C_w	
	Global-SSV	Local-SSV
$\mathcal{V}_{\boldsymbol{G}}$	6.6772E-04	
$\mathcal{V}_{\boldsymbol{G},C_w}$	6.5408E-04	
$\mathcal{V}_{\boldsymbol{G},\boldsymbol{MI}^2}$	6.7591E-04	6.6582E-04
$\mathcal{V}_{\boldsymbol{G},\boldsymbol{MI}^3}$	6.6895E-04	**6.1056E-04**
$\mathcal{V}_{\boldsymbol{G},\boldsymbol{MI}^4}$	6.8511E-04	6.7065E-04
$\mathcal{V}_{\boldsymbol{G},\boldsymbol{MI}^{2,3}}$	7.0910E-04	6.9833E-04
$\mathcal{V}_{\boldsymbol{G},\boldsymbol{MI}^{2,4}}$	6.9694E-04	6.8407E-04
$\mathcal{V}_{\boldsymbol{G},\boldsymbol{MI}^{3,4}}$	7.0599E-04	6.9029E-04
$\mathcal{V}_{\boldsymbol{G},\boldsymbol{MI}^{2,3,4}}$	6.9875E-04	6.8052E-04

Table 6.3 provides the average C_w values obtained at the final iteration over three runs for all cases. It can be seen that all cases show a substantial improvement when compared to the parent design whose C_w value is 1.025×10^{-3}. However, there is no significant difference between optimum designs generated from $\mathcal{V}_{\boldsymbol{G}}$, $\mathcal{V}_{\boldsymbol{G},C_w}$ and the shape-supervised subspaces. When comparing global- to composite-SSVs, the latter perform consistently better. However, slightly, and the overall best is achieved by $\mathcal{V}_{\boldsymbol{G},\boldsymbol{MI}^3}$ using a composite-SSV, which might be a rather unexpected result since it surpasses the performance attained by $\mathcal{V}_{\boldsymbol{G},C_w}$.

6.4.6 Computational cost

The computational cost of constructing shape-supervised subspaces is higher than that of a purely grid-based geometric subspace. However, using moments is glaringly cheaper than performing physics simulations, in our case, C_w. On a PC with a dual 24-core 2.7GHz Intel® Xeon® Gold 6226 CPU and 128GB of memory, it takes approximately 9.04 seconds to evaluate all moment invariants $\{\boldsymbol{MI}^s, s = 1, 2, 3, 4\}$ for a hull meshed with 2,512,886 vertices. On the other hand, the employed potential flow solver requires approximately 69.30 seconds for a single evaluation of C_w for a hull meshed with $2,250$ vertices. Therefore, performing dimension reduction with shape-supervised approaches provides the same (or better) quality with a significantly lower computational cost when compared to the other supervised techniques in this work.

6.5 Conclusion and future work

Despite the success of design space dimensionality reduction for accelerating computationally demanding shape optimisation processes, the existing approaches suffer from two critical drawbacks: i) low-levels of robustness, i.e., a non-negligible percentage of designs in the reduced dimensionality subspace corresponds to invalid/infeasible instances, and ii) inability to capture high-level structure information, i.e., high-level features, associated to physics, which would considerably improve performance, are not captured. Therefore, in this work, we propose a shape-supervised approach for reducing the dimension of the initial design space. Our approach uses geometric moment invariants of both global and composite nature to construct a shape-signature vector (SSV) that describes important underlying intrinsic structures of the shape, which can, to some extent, substitute physics information. The subspaces produced in this work retain the required reconstruction capabilities, offer diversity and robustness, and, more importantly, are physics informed. The representation capacity and compactness of the produced subspaces are accessed, and the former is equivalent to the original spaces. In contrast, the latter is significantly better,

i.e., fewer invalid designs are generated.

Furthermore, the applicability of the proposed method is tested against the challenging problems of wing design and ship-hull shape optimisation. The wing and hull models are parameterised with 33 and 27 design variables parameters, respectively. The shape optimisation performed for the hull model aims at its wave resistance coefficient (C_w) minimisation. The results confirm our claims and demonstrate the higher convergence capability of the shape-supervised approach. One may easily apply the same approach to shape optimisation of other free-form shapes in computational mechanics.

In the future, we would also like to explore the possibility of SSV's integration into a generative adversarial network and perform physics-augmented training. At the same time, an extension of our work in an Iso-Geometric Analysis setting, where Non-uniform Rational B-splines representations (NURBS) of the shape for DSDR, analysis and shape optimisation would be directly used, is also in our plans.

Chapter 7

Physics-informed feature-to-feature learning

7.1 Introduction

As has been discussed so far in this thesis, data-driven meta-modelling is used to release the computational burden of SDD, and dimensionality reduction, [152] and dimensionality reduction [22] 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, also referred to as a feature or manifold learning [6], creating a lower-dimensional latent representation of the original space. In a design context, these techniques are structured on the assumption that the geometric variability in design space is not the same in all directions. Only a few inherent feature directions materialise most improvement in the design. These inherent features can form the basis of a new lower-dimensional latent subspace [153]. 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 [54]; (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, a 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 the design's physics might be different [154, 155]. Therefore, it is essential, especially in the context of surrogate modelling, that the information about QoI should be present during the feature extraction. Hence, the latent space includes both geometric and functional variability. As discussed earlier, to tackle this problem, the Active Subspace Method (ASM) was proposed by Lukaczyk et al. [20] and Constantine [79], 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 possible, to calculate accurately. These gradients can be locally approximated using different techniques, such as the 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 high-dimensional problems. Moreover, the subspace created with ASM is solely formulated with gradients of QoI. Hence, capturing the geometric variability of the original design space may take time. As the design's geometric variance and its corresponding variability in QoI is interdependent, one could first extract the geometric features and then the functional features.

The technique proposed in this chapter is formulated around a dimension-reduction pipeline that extracts the important features of a given design space first in terms of the geometric variability of designs and then in terms of the variability of QoI. We refer to 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 of 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 designs 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 terms of accuracy surrogate modelling and optimisation results. The sequential layout of the proposed approach is illustrated in Fig. 7.1. The code of this pipeline is available at `https://github.com/shahrozkhan66/PIFFL.git`.

The remainder of this chapter is organised as follows: Section 7.2 gives comprehensive details on the formulation of the proposed approach. The numerical results to prove the working and feasibility of the proposed technique are presented in Section 7.3. The concluding remarks and opportunities for future work are included in Section Section 7.4.

7.2 Proposed methodology

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

7.2.1 General definitions and assumptions

Let a geometric domain $\mathcal{G}$ representing a baseline design and a set of coordinates $\boldsymbol{\zeta} \in \mathcal{G} \subset \mathbb{R}^m$ with $m = 1, 2, 3$. Now also assume a design parameter vector $\mathbf{x} = \{x_k, k = 1, 2, \ldots, M\} \in \mathcal{X} \subset \mathbb{R}^M$, where $\mathcal{X}$ is subset of $\mathbb{R}^M$ and is bounded with lower ($\mathbf{x}^l$) and upper

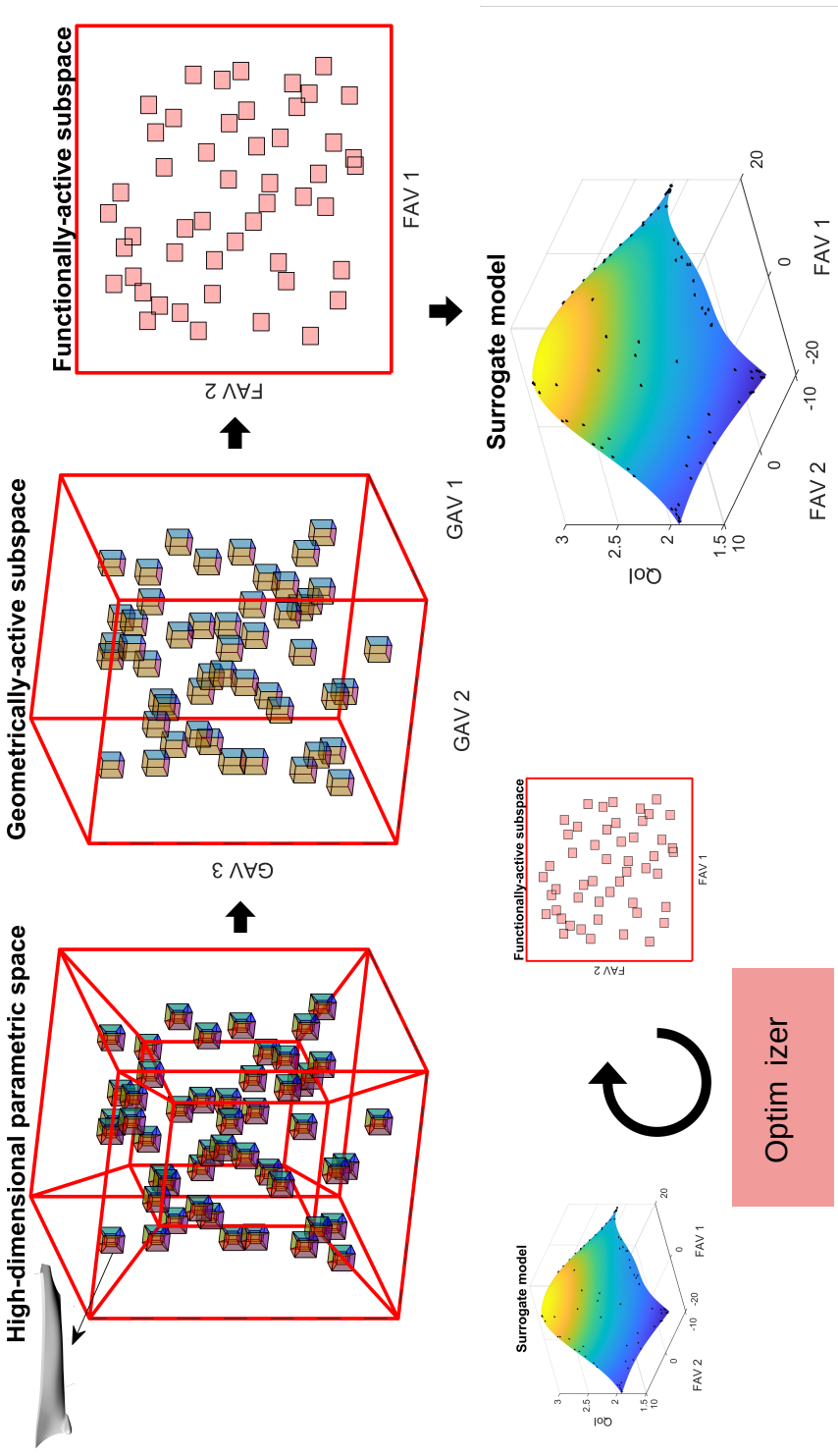


Figure 7.1: Sequential workflow of the proposed approach, which commences with high-dimensional design space, represented with a hypercube, having a large feasible variation of the baseline design. Afterwards, geometric features of this space are identified to span the basis of a new geometrically-active subspace. This subspace is then used to construct a functionally-active subspace to reduce dimensionality further and accumulate geometric and functional variability. After that, the surrogate model is developed with designs sampled from this subspace and explored with an optimiser to find an optimal design.

($\mathbf{x}^u$) geometric bounds creating a viable $M-$dimensional design space. The associated parametrisation of $\mathcal{G}$ defines a geometric modification vector $\boldsymbol{\delta}(\boldsymbol{\zeta}, \mathbf{x}) \in \mathbb{R}^n$, with $n = 1, 2, 3$, which for $\mathbf{x} \in \mathcal{X}$ modifies each $\boldsymbol{\zeta} \in \mathcal{G}$ of the baseline shape to new geometry $\boldsymbol{\zeta}' \in \mathcal{G}'$ as:

$$\boldsymbol{\zeta}' = \boldsymbol{\zeta} + \boldsymbol{\delta}(\boldsymbol{\zeta}, \mathbf{x}), \tag{7.1}$$

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 $\boldsymbol{\delta}(\boldsymbol{\zeta}, \mathbf{x})$ first in term of its geometric variability, $\hat{\boldsymbol{\delta}}(\boldsymbol{\zeta}, \mathbf{v})$, and then take this representation to find another reduced representation, $\hat{\boldsymbol{\delta}}(\boldsymbol{\zeta}, \mathbf{u})$, in terms of functional variability of $\mathcal{G}$ with respect to QoI such as drag, resistance, stability, etc. Here, $\mathbf{v} = \{v_i, i = 1, 2, \ldots, N\} \in \mathcal{V} \subset \mathbb{R}^N$ 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 $\mathbf{u} = \{u_i, i = 1, 2, \ldots, P\} \in \mathcal{U} \subset \mathbb{R}^P$ 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. 7.2 graphically illustrators shows the notation set for the formulation of the proposed approach at $m = 3$, $n = 3$, $M = 13$ and $N = 7$.

The objective is to minimise an objective/cost function defining the reconstruction error between the original variables and their lower-dimensional representation. In the present case, this error measures the accuracy of $\hat{\boldsymbol{\delta}}(\boldsymbol{\zeta}, \mathbf{v})$ and $\hat{\boldsymbol{\delta}}(\boldsymbol{\zeta}, \mathbf{u})$ in term of Mean Squared Error (MSE) normalised to total geometric (σ_g^2) and functional (σ_f^2) variance as in Eq. (7.2) and (7.3), respectively.

$$NMSE_g = \frac{MSE_g}{\sigma_g^2} = \frac{\iint_{\mathcal{X}\times\mathcal{V},\mathcal{G}} ||\boldsymbol{\delta}(\boldsymbol{\zeta}, \mathbf{x}) - \hat{\boldsymbol{\delta}}(\boldsymbol{\zeta}, \mathbf{v})||^2 \rho(\mathbf{x}, \mathbf{v}) d\boldsymbol{\zeta} d\mathbf{x} d\mathbf{v}}{\iint_{\mathcal{X},\mathcal{G}} ||\boldsymbol{\delta}(\boldsymbol{\zeta}, \mathbf{x})||^2 \rho(\mathbf{x}) d\boldsymbol{\zeta} d\mathbf{x}} \tag{7.2}$$

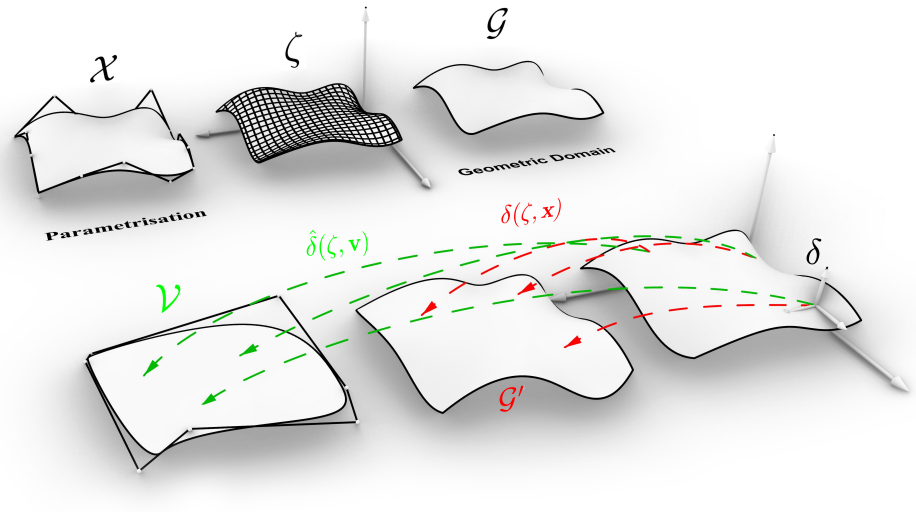


Figure 7.2: Representation of the scheme and notation used for the current formulation.

$$NMSE_f = \frac{MSE_f}{\sigma_f^2} = \frac{\iint_{\mathcal{V}\times\mathcal{U},\mathcal{G}} ||\hat{\boldsymbol{\delta}}(\boldsymbol{\zeta}, \mathbf{v}) - \hat{\boldsymbol{\delta}}(\boldsymbol{\zeta}, \mathbf{u})||^2 \rho(\mathbf{v}, \mathbf{u}) d\boldsymbol{\zeta} d\mathbf{v} d\mathbf{u}}{\iint_{\mathcal{V},\mathcal{G}} ||\boldsymbol{\delta}(\boldsymbol{\zeta}, \mathbf{v})||^2 \rho(\mathbf{v}) d\boldsymbol{\zeta} d\mathbf{v}} \tag{7.3}$$

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

7.2.2 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, $\mathcal{C} \in \{\mathcal{C}^g, \mathcal{C}^f\}$, where $\mathcal{C}^g$ and $\mathcal{C}^f$ are the covariance matrix for geometric and functional feature extraction, respectively. Once $\mathcal{C}$ is constructed its eigendecomposition follows the same procedure for both geometric and functional feature extraction. Therefore, in the subsequent sections, we explain the eigendecomposition and then describe the formulation of $\mathcal{C}^g$ and $\mathcal{C}^f$.

$\mathcal{C}$ is a symmetric and positive definite matrix and constitutes of elements of ambient or auxiliary space, $\mathcal{Z} \in \{\mathcal{X}, \mathcal{V}\} \subset \mathbb{R}^M$. For learning geometric features, these elements are the associated shape parameters. For functional features, these elements are the gradients

of QoI with respect to the dimensional coordinates, $\mathbf{z}$, of $\mathcal{Z}$. The eigendecomposition of $\mathcal{C}$ for $\mathcal{Z}$ gives:

$$\mathcal{C} = \mathbf{W}\mathbf{\Lambda}\mathbf{W}^T. \tag{7.4}$$

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

$$\mathbf{\Lambda} = \begin{bmatrix} \mathbf{\Lambda}_1 & \\ & \mathbf{\Lambda}_2 \end{bmatrix}, \quad \mathbf{W} = \begin{bmatrix} \mathbf{W}_1 & \mathbf{W}_2 \end{bmatrix}. \tag{7.5}$$

The columns of $\mathbf{W}_1 = \{\mathbf{w}_i, i = 1, 2, \ldots, N\}$ and $\mathbf{W}_2 = \{\mathbf{w}_j, j = 1, 2, \ldots, M - N\}$ span the dominant and non-dominant features of $\mathcal{Z}$, which defines the active and inactive subspaces, receptively, and $\mathbf{\Lambda}_1 = \{\lambda_i, i = 1, 2, \ldots N\}$ and $\mathbf{\Lambda}_2 = \{\lambda_j, j = 1, 2, \ldots M - N\}$ 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 natures, GAS and FAS. Therefore, the decision on a number of active and inactive features for GAS and FAS is made 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, $\mathbf{\Lambda}_1 = \{\lambda_i, i = 1, 2, \ldots N\}$. Afterwards, $\mathbf{z}$ can be project on these subspaces using Eq. (7.6) to find its active $\mathbf{a}^1 = \{a_i^1, i = 1, 2, \ldots, N\}$ and inactive $\mathbf{a}^2 = \{a_j^2, j = 1, 2, \ldots, M - N\}$ latent parameters.

$$\mathbf{a}^1 = \mathbf{W}_1^T \mathbf{z} \in \mathbb{R}^N, \quad \mathbf{a}^2 = \mathbf{W}_2^T \mathbf{z} \in \mathbb{R}^{M-N}. \tag{7.6}$$

Among $\mathbf{a}^1$ and $\mathbf{a}^2$, we are only interested in $\mathbf{a}^1$ as its basis $\mathbf{W}_1$ covers the 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, $a_i^1 = w_{1,i}z_1 + w_{2,i}z_2 + \cdots + w_{M,i}z_M$ ($a_j^2 = w_{1,j}z_1 + w_{2,j}z_2 + \cdots + w_{M,j}z_M$).

For most practical problems, evaluation of $\mathcal{C}$ 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 the one studied in the present work, the estimation of $\mathcal{C}$, $\hat{\mathcal{C}}$, is evaluated using pseudo-random sampling techniques such as Monte Carlo or Latin hypercube sampling. Now the eigendecomposition of $\hat{\mathcal{C}}$ gives estimated eigenpairs, $\hat{\mathbf{W}}$ and $\hat{\mathbf{\Lambda}}$:

$$\mathcal{C} \approx \hat{\mathcal{C}} = \hat{\mathbf{W}}\hat{\mathbf{\Lambda}}\hat{\mathbf{W}}^T. \tag{7.7}$$

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

$$\omega = \text{dist}(\text{ran}(\mathbf{W}_1), \text{ran}(\hat{\mathbf{W}}_\mathbf{1})) = \left|\left|\mathbf{W}_1\mathbf{W}_1^T - \hat{\mathbf{W}}_1\hat{\mathbf{W}}_1^T\right|\right|. \tag{7.8}$$

7.2.3 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, $\mathcal{C}_g$. 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 $\boldsymbol{\delta}(\boldsymbol{\zeta}, \mathbf{x})$ as:

$$\mathbf{d}(\mathbf{x}) = \begin{bmatrix} \mathbf{d}_1(\mathbf{x}) \\ \mathbf{d}_2(\mathbf{x}) \\ \mathbf{d}_3(\mathbf{x}) \end{bmatrix} = \begin{bmatrix} d_{1,1}(\mathbf{x}) \\ \vdots \\ d_{1,E}(\mathbf{x}) \\ d_{2,1}(\mathbf{x}) \\ \vdots \\ d_{2,E}(\mathbf{x}) \\ \vdots \\ d_{3,1}(\mathbf{x}) \\ \vdots \\ d_{3,E}(\mathbf{x}) \end{bmatrix}, \tag{7.9}$$

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

$$\mathcal{C}^g = \int_{\mathcal{X}} \mathbf{d}(\mathbf{x})\mathbf{d}(\mathbf{x})^T \rho(\mathbf{x}) d\mathbf{x}. \tag{7.10}$$

As explained earlier, the analytical solution of C^g requires solving high-dimensional integral. Therefore, its approximation is obtained as in Eq. (7.11) while sampling $\mathcal{X}$ with a statistically convergent number of Monte Carlo realisations, S_g, which creates dataset $\mathbf{X} = \{\mathbf{x}_r, r = 1, 2, \ldots, S_g\} \sim \rho(\mathbf{x})$.

$$\mathcal{C}^g = \hat{\mathcal{C}}^g = \frac{1}{S_g} \sum_{r=1}^{S_g} \mathbf{d}(\mathbf{x}_r)\mathbf{d}(\mathbf{x}_r)^T = \hat{\mathbf{W}}^g \hat{\mathbf{\Lambda}}^g \left(\hat{\mathbf{W}}^g\right)^T \approx \mathbf{W}^g \mathbf{\Lambda}^g \mathbf{W}^{gT}. \tag{7.11}$$

Here, $\mathbf{W}^g$ is $[L \times L]$ matrix whose columns are the eigenvectors. The eigenvalues Λ obtained from the eigendecomposition of $\mathcal{C}$ represent 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:

$$\sigma^2 = \frac{\sum_{i=1}^{N} \lambda_i}{\sum_{i=1}^{M} \lambda_i}, \tag{7.12}$$

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:

$$\mathbf{v} = \mathbf{W}_1^{gT} \mathbf{d}(\mathbf{x}) \in \mathcal{V} \subset \mathbb{R}^N. \tag{7.13}$$

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. (7.2) can also be expressed in discrete form as:

$$MSE_g = \frac{1}{S_g} \sum_{r=1}^{S_g} \left|\left| \hat{\mathbf{d}}(\mathbf{v}_r) - \mathbf{d}(\mathbf{x}_r) \right|\right|^2, \tag{7.14}$$

where

$$\sigma_g^2 = \frac{1}{S_g} \sum_{r=1}^{S_g} \left|\left| \mathbf{d}(\mathbf{x}_r) \right|\right|^2. \tag{7.15}$$

Combining equation Eq. (7.14) and Eq. (7.15) gives NMSE as:

$$NMSE_g = \frac{\sum_{r=1}^{S_g} \left|\left| \hat{\mathbf{d}}(\mathbf{v}_r) - \mathbf{d}(\mathbf{x}_r) \right|\right|^2}{\sum_{r=1}^{S_g} \left|\left| \mathbf{d}(\mathbf{x}_r) \right|\right|^2}. \tag{7.16}$$

Physics-informed geometrically-active subspace

Based on [155], the geometric features can also be extracted containing variability of both geometry and physics. We refer to 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, $\boldsymbol{\delta} \in \mathbb{R}^n$, there is a lumped (or global) physical parameter vector, $\varepsilon \in \mathbb{R}^{n^*}$ with $n^* = 1, \ldots, \infty$, 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, $\boldsymbol{\zeta}_\theta$, 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 $\gamma(\boldsymbol{\zeta}, \mathbf{x}) \in \mathbb{R}^n$ with $n = max\{n, n^*\}$ as:

$$\boldsymbol{\gamma}(\boldsymbol{\zeta}, \mathbf{x}) = \begin{cases} \boldsymbol{\delta}(\boldsymbol{\zeta}, \mathbf{x}) & \text{if } \boldsymbol{\zeta} \in \mathcal{G} \\ \varepsilon(\boldsymbol{\zeta}, \mathbf{x}) & \text{if } \boldsymbol{\zeta} \in \mathcal{P} \end{cases} \tag{7.17}$$

Similar to GAS, here, the dimensionality reduction aims to identify a reduced-dimensionality representation $\hat{\boldsymbol{\gamma}}(\boldsymbol{\zeta}, \mathbf{v}^*)$. Here, $\mathbf{W}_1^{g*}$ is a *physics-informed geometric feature* set of $\mathcal{X}$, which is obtained with eigendecomposition of covariance matrix $\mathcal{C}^{g^*}$ given in Eq. (7.18). The linear combination of $\mathbf{W}_1^{g*}$ with the discretisation of $\boldsymbol{\gamma}(\boldsymbol{\zeta}, \mathbf{x})$, $\mathbf{d}(\mathbf{x}^*)$, 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. (7.19)).

$$\mathcal{C}^{g^*} = \int_{\mathcal{G}} \mathbf{d}(\mathbf{x}^*)\mathbf{d}(\mathbf{x}^*)^T \rho(\mathbf{x}) d\mathbf{x}, \tag{7.18}$$

$$\mathbf{v}^* = \mathbf{W}_1^{g*T}\mathbf{d}(\mathbf{x}^*) \in \mathcal{V}^* \subset \mathbb{R}^{N^*}. \tag{7.19}$$

7.2.4 Functionally-active subspace

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

$$f = f(\mathbf{v}), \quad \nabla_{\mathbf{v}} f = \nabla_{\mathbf{v}} f(\mathbf{v}) = \begin{bmatrix} \frac{\partial f}{\partial v_1}(\mathbf{v}) \\ \vdots \\ \frac{\partial f}{\partial v_N}(\mathbf{v}) \end{bmatrix}, \quad \text{for all } \mathbf{v} \in \mathcal{V}. \tag{7.20}$$

The objective 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 $\mathcal{C}^f$ (given in Eq. (7.21)), which is defined as an average of the outer product of $\nabla_{\mathbf{v}} f$ with itself:

$$\mathcal{C}^f = \int_{\mathcal{V}} (\nabla_{\mathbf{v}} f)(\nabla_{\mathbf{v}} f)^T \rho(\mathbf{v}) d\mathbf{v}. \tag{7.21}$$

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. (7.21), which is obtained using the following scheme:

$$\mathcal{C}^f \approx \hat{\mathcal{C}}^f = \frac{1}{S_f} \sum_{r=1}^{S_f} \nabla_{\mathbf{v}} f(\mathbf{v}_r)\ \nabla_{\mathbf{v}} f(\mathbf{v}_r)^T = \hat{\mathbf{W}}^f \hat{\mathbf{\Lambda}}^f \hat{\mathbf{W}}^{f^T} \approx \mathbf{W}^f \mathbf{\Lambda}^f \mathbf{W}^{f^T}, \tag{7.22}$$

where S_f is the number of designs points sampled from $\mathcal{V}$ and the accuracy of $\hat{\mathcal{C}}^f$ 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 ($\mathbf{W}_2^f$) feature is done by looking for gaps in the elements of $\mathbf{\Lambda}$. 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 the 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 $\mathbf{\Lambda}_2^f \approx 0$ (i.e., $\{\lambda_{P+1}, \lambda_{P+2}, \ldots, \lambda_M\} \approx 0$) 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:

$$\mathbf{u} = \mathbf{W}_1^{f^T}\mathbf{v} = \mathbf{W}_1^{f^T}\left(\mathbf{W}_1^{g^T}\mathbf{d}(\mathbf{x})\right) \in \mathcal{U} \subset \mathbb{R}^P. \tag{7.23}$$

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

$$\hat{\mathbf{v}} = \mathbf{W}_1^f\mathbf{u} = \mathbf{W}_1^f\mathbf{W}_1^{f^T}\mathbf{v}, \tag{7.24}$$

where variable $\mathbf{u}$ defines the reduced dimensionality representation $\hat{\boldsymbol{\delta}}(\boldsymbol{\zeta}, \mathbf{u})$ of shape modification vector $\hat{\boldsymbol{\delta}}(\boldsymbol{\zeta}, \mathbf{v})$ in functionally-active subspace, which is the reduced dimensional geometrically-active representation of shape modification vector $\boldsymbol{\delta}(\boldsymbol{\zeta}, \mathbf{x})$ 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

$$NMSE_f = \frac{\sum_{r=1}^{S_f}\left|\left|\hat{\mathbf{d}}(\mathbf{u}_r) - \mathbf{d}(\mathbf{x}_r)\right|\right|^2}{\sum_{r=1}^{S_f}\left|\left|\mathbf{d}(\mathbf{x}_r)\right|\right|^2}. \tag{7.25}$$

Similarly, the FAS, $\mathcal{U}^*$, of $\mathcal{V}^*$ is obtained after replacing $\mathbf{v}$ with $\mathbf{v}^*$ in Eq. (7.22). The basis of $\mathcal{U}^*$ is defined by a latent variable $\mathbf{u}^*$, which is again composed as a linear combination of $\mathbf{v}^*$ with physics-informed functional features $\mathbf{W}_1^{f^*}$.

7.2.5 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 [20]. Therefore, by 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

$$f(\mathbf{v}) \approx g(\mathbf{W}_1^{f^T}\mathbf{v}) = g(\mathbf{u}) \tag{7.26}$$

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

$$g(\mathbf{u}) \approx g^*(\mathbf{u}) \equiv \mathcal{R}(\mathbf{u}; g_1, g_2, \ldots, g_{S_r}). \tag{7.27}$$

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

$$\mathcal{U} = \{\mathbf{u} = \mathbf{W}_1^{f^T}\mathbf{v},\ \mathbf{v} \in \mathcal{V}\} \subset \mathbb{R}^P. \tag{7.28}$$

This work uses different surrogate modelling techniques, 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 7.3.3. We have seen better results in training the surrogate model with GPR. Thus, leaving the comprehensive details to [156, 157], the below section gives brief deception of the critical concepts and construction of the GPR-based surrogate model.

Gaussian process regression

GPR is a non-parametric Bayesian approach [156], which has been used in different design applications [158]. 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 the relationship between inputs and outputs with small data size, (2) easily handle noise in the data, thus, avoiding 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:

$$y = g(\mathbf{u}) + \epsilon, \text{ with } \epsilon \sim \mathcal{N}(0, \sigma_\epsilon^2). \tag{7.29}$$

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 [157]. 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):

$$g(\mathbf{u}) \sim GP(\mu(\mathbf{u}), k(\mathbf{u}, \mathbf{u}')). \tag{7.30}$$

A GP is any distribution over functions such that any finite set of function values $g(\mathbf{u}_1), g(\mathbf{u}_2), \ldots, g(\mathbf{u}_{S_r})$ have a joint Gaussian distribution and is defined by a mean $\mu(\mathbf{u})$. A covariance $k(\mathbf{u}, \mathbf{u}')$ function [157]. The $\mu(\mathbf{u})$ defines the expected function value at input $\mathbf{u}$,

$$\mu(\mathbf{u}) = \mathbb{E}[g(\mathbf{u})], \tag{7.31}$$

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. (7.30) defines dependence between the $g(\mathbf{u})$ and $g(\mathbf{u}')$ as:

$$k(\mathbf{u}, \mathbf{u}') = Cov\left[g(\mathbf{u}), g(\mathbf{u}')\right] = \mathbb{E}\left[\{g(\mathbf{u}) - \mu(\mathbf{u})\}\{g(\mathbf{u}') - \mu(\mathbf{u}')\}\right]. \tag{7.32}$$

Once $\mu(\mathbf{x})$ and $k(\mathbf{u}, \mathbf{u}')$ are chosen, we can use GP to draw prior 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;

$$\mathcal{D}^t = \{\mathbf{U}^t, \mathbf{g}^t\}, \tag{7.33}$$

where

$$\mathbf{U}^t = \begin{bmatrix} \mathbf{u}_1^t \\ \mathbf{u}_1^t \\ \vdots \\ \mathbf{u}_{S_r}^t \end{bmatrix} = \begin{bmatrix} u_{1,1}^t & u_{1,2}^t & \dots & u_{1,N}^t \\ u_{2,1}^t & u_{2,2}^t & \dots & u_{2,N}^t \\ \vdots & \vdots & \ddots & \vdots \\ u_{S_r,1}^t & u_{S_r,2}^t & \dots & u_{S_r,N}^t \end{bmatrix}, \quad \mathbf{g}^t = \begin{bmatrix} g(\mathbf{u}_1^t) \\ g(\mathbf{u}_2^t) \\ \vdots \\ g(\mathbf{u}_{S_r}^t) \end{bmatrix}. \tag{7.34}$$

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 $\mathbf{g}^*$ follow a joint multivariate normal distribution, which can be written as follows:

$$\begin{bmatrix} \mathbf{g}^t \\ \mathbf{g}' \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} K(\mathbf{U}^t, \mathbf{U}^t) + \sigma_\epsilon^2 \mathbf{I} & K(\mathbf{U}^t, \mathbf{U}') \\ K(\mathbf{U}', \mathbf{U}^t) & K(\mathbf{U}', \mathbf{U}') \end{bmatrix}\right). \tag{7.35}$$

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

$$K(\mathbf{U}^t, \mathbf{U}^t) = \begin{bmatrix} k(\mathbf{u}_1^t, \mathbf{u}_1^t) & k(\mathbf{u}_1^t, \mathbf{u}_2^t) & \dots & k(\mathbf{u}_1^t, \mathbf{u}_{S_r}^t) \\ k(\mathbf{u}_2^t, \mathbf{u}_1^t) & k(\mathbf{u}_2^t, \mathbf{u}_2^t) & \dots & k(\mathbf{u}_2^t, \mathbf{u}_{S_r}^t) \\ \vdots & \vdots & \ddots & \vdots \\ k(\mathbf{u}_{S_r}^t, \mathbf{u}_1^t) & k(\mathbf{u}_{S_r}^t, \mathbf{u}_2^t) & \dots & k(\mathbf{u}_{S_r}^t, \mathbf{u}_{S_r}^t) \end{bmatrix}, \tag{7.36}$$

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. (7.35), parameters $\mathbf{I}$ and σ_ϵ^2 are the identity matrix and noise level of observations, respectively. Following [157], the conditional distribution $\rho(\mathbf{g}'|\mathbf{U}^t, \mathbf{y}^t, \mathbf{U}^t)$ is then a multivariate normal distribution with mean (Eq. (7.37)) and covariance matrix (Eq. (7.38)).

$$\mu^t(\mathbf{u}) = K(\mathbf{u}, \mathbf{U}^t)\left[K(\mathbf{U}^t, \mathbf{U}^t) + \sigma_\epsilon^2 \mathbf{I}\right]^{-1} \mathbf{g}^t, \tag{7.37}$$

$$k_t(\mathbf{u}, \mathbf{u}') = k(\mathbf{u}, \mathbf{u}') - K(\mathbf{u}, \mathbf{U}^t) \left[K(\mathbf{U}^t, \mathbf{U}^t) + \sigma_\epsilon^2 \mathbf{I} \right]^{-1} K(\mathbf{U}^t, \mathbf{u}'). \tag{7.38}$$

In GPR, k is known as the *kernel* function [156], and a suitable choice of this function is based on assumptions such as smoothness and likely patterns to be expected in the training data [157]. For GPR, different types of kernels have been developed and used. However, most common is the *exponential kernel function*,

$$k(\mathbf{u}, \mathbf{u}') = \sigma_s^2 \exp\left(\frac{-||\mathbf{u} - \mathbf{u}'||^2}{2\sigma_l^2} \right), \tag{7.39}$$

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 prior 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 [156], or Bayesian techniques [159], which measures how well a given θ describes $\mathcal{D}$. This work tested GPR with different kernel functions, and hyper-parameters were optimised with the Bayesian technique [159].

7.2.6 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 is evaluated based on heuristic presented in [20] and is given in Eq. (7.40).

$$\begin{aligned} \mathbf{u}^l &= \mathbf{x}^l \times \text{diag}(\text{sign}(\mathbf{W}_1^f)^T \mathbf{W}_1^f), \\ \mathbf{u}^u &= \mathbf{x}^u \times \text{diag}(\text{sign}(\mathbf{W}_1^f)^T \mathbf{W}_1^f), \end{aligned} \tag{7.40}$$

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:

$$\begin{aligned} \text{given} \quad & \mathbf{u} \in \mathbb{R}^P \\ \min_{\mathbf{u}} \quad & g(\mathbf{u}) \\ \text{subject to} \quad & \mathbf{u} \in \mathcal{U} \\ \text{yield} \quad & \mathbf{u}_{optimal}, \end{aligned} \tag{7.41}$$

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 $\mathcal{X}$. 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:

$$\begin{aligned} \text{given} \quad & \mathbf{u} = \mathbf{u}_{optimal} \\ & \hat{\mathbf{d}}(\mathbf{u}) = \mathbf{W}_1^g \mathbf{W}_1^f \mathbf{u}, \quad \mathbf{W}_1^g \in \mathbb{R}^{L \times N}, \quad \mathbf{W}_1^f \in \mathbb{R}^{N \times P} \\ & \mathbf{d}(\mathbf{x}_r) \text{ of } \boldsymbol{\delta}(\boldsymbol{\zeta}, \mathbf{x}_r) \text{ for } r^{th} \text{ design} \\ \min_{\mathbf{x}} \quad & \tau_1 \times g(\mathbf{W}_1^{fT}(\mathbf{W}_1^{gT}\mathbf{d}(\mathbf{x}))) + \tau_2 \times ||\hat{\mathbf{d}}(\mathbf{u}) - \mathbf{d}(\mathbf{x})||^2 \\ \text{subject to} \quad & g(\mathbf{W}_1^{fT}(\mathbf{W}_1^{gT}\mathbf{d}(\mathbf{x}))) \leq g(\mathbf{u}) \\ & \mathbf{x}^{\mathbf{l}} \leq \mathbf{x} \leq \mathbf{x}^{\mathbf{u}} \\ \text{yield} \quad & \mathbf{x}_{optimal} \end{aligned} \tag{7.42}$$

The term $||.||$ 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. (7.41), we have added $g(.)$ in the objective function of Eq. (7.42) to find if there is any further improvement that can be achieved during the exploration of $\mathcal{X}$ for $\mathbf{x}_{optimal}$. τ_1 and τ_2 adjust

the weight of $g(.)$ and $||.||$ 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 $||.||$ by setting $\tau_2 > \tau_1$. The constraint $g(\mathbf{W}_1^{f^T}(\mathbf{W}_1^{g^T}\mathbf{d}(\mathbf{x}))) \leq g(\mathbf{u})$ in Eq. (7.42) ensures that $\mathbf{x}_{optimal}$ has either the same or higher performance than $\mathbf{u}_{optimal}$. Algorithm 4 gives the stepwise procedure for implementing the proposed approach.

7.3 Results and discussion

To validate feature-to-feature learning, we tested three different SDO pipelines on the DTMB Naval Ship hull shown in Fig, 5.2 (b) under the same setting as described in §5.3.2 and 5.3.3. The sequential workflow of these pipelines is shown in Fig. 7.3. The first pipeline is the conventional ASM, which learns the lower dimensional FAS, $\mathcal{U}_1$, from gradients of c_w ($\Delta_{\mathbf{x}} c_w$) evaluated with respect to design parameters. The second pipeline is based on the proposed two-step feature-to-feature learning. First, a GAS representation of the original design space is learned in this pipeline. Then, functional features of this subspace are extracted with eigendecomposition of the covariance matrix composed of gradients ($\Delta_{\mathbf{v}} c_w$), 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, containing the variability of 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 the optimiser to find an optimal design for the test case. The efficiency of these pipelines is evaluated in terms of the 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

Algorithm 4 Step wise procedure of the proposed feature-to-feature learning approach.

1: Create an initial model $\mathcal{G}$ composed of coordinate set $\boldsymbol{\zeta} \in \mathcal{G} \subset \mathbb{R}^m$ and parameterise it with M design parameters $(x_1, x_2, \ldots, x_M)$.
2: Define the design space $\mathcal{X}$ with lower and upper bounds of M parameters, $\mathcal{X} := \{x_k^l \leq x_k \leq x_k^u,\ \forall k \in \{1, 2, \ldots M\}\}$.
3: Define a shape modification vector $\boldsymbol{\delta}(\boldsymbol{\zeta}, \mathbf{x})$ to modify $\mathcal{G}$ for any realisation $\mathbf{x} \in \mathcal{X}$.
4: Sample $\mathcal{X}$ to create set $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{S_g}]^T$, where S_g is sample size.
5: Evaluate shape modification set $\boldsymbol{\delta}(\boldsymbol{\zeta}, \mathbf{X}) = [\boldsymbol{\delta}(\boldsymbol{\zeta}, \mathbf{x}_1), \boldsymbol{\delta}(\boldsymbol{\zeta}, \mathbf{x}_2), \ldots, \boldsymbol{\delta}(\boldsymbol{\zeta}, \mathbf{x}_{S_g})]^T$
6: Create the dataset $\mathbf{D}(\mathbf{X})$ containing discrete representation of $\boldsymbol{\delta}(\boldsymbol{\zeta}, \mathbf{X})$ as $\mathbb{D}(\mathbf{X}) = [\mathbf{d}(\mathbf{x}_1), \mathbf{d}(\mathbf{x}_2), \ldots, \mathbf{d}(\mathbf{x}_{S_g})]^T$, where the discretisation of i^{th} design $\mathbf{x}_i$ is represented in Eq. (7.9).
7: Compute $\mathcal{C}_g$ and its eigendecomposition: $\mathcal{C}_g = \frac{1}{S_g} \sum_{r=1}^{S_g} \mathbf{d}(\mathbf{x}_r)\mathbf{d}(\mathbf{x}_r)^T = \mathbf{W}^g \boldsymbol{\Lambda}^g \mathbf{W}^{gT}$
8: Partition eigenspace of $\mathcal{C}_g$ as in Eq. (7.5), where elements of $\boldsymbol{\Lambda}_1^g$ 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 $\mathbf{v} = \mathbf{W}_1^{gT} \mathbf{d}(\mathbf{x})$, $\mathbf{v} = \{v_r, r = 1, 2, \ldots, N\}$.
10: With $\mathbf{v}$ create a geometrically-active subspace $\mathcal{V} \subset \mathbb{R}^N$ as, $\mathcal{V} := \{v_k^l \leq v_k \leq v_k^u,\ \forall k \in \{1, 2, \ldots N\}\}$, where v_k^l and v_k^u are the lower and upper bounds and $M < N$.
11: Sample $\mathcal{V}$ to create set $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_{S_f}]^T$, where S_f is sample size.
12: Evaluate QoI for $\mathbf{V}$, $\mathbf{F} = [f(\mathbf{v}_1), f(\mathbf{v}_2), \ldots, f(\mathbf{v}_N)]^T$
13: Evaluate Gradients $\nabla \mathbf{F} = [\nabla_{\mathbf{v}} f(\mathbf{v_1}), \nabla_{\mathbf{v}} f(\mathbf{v_2}), \ldots, \nabla_{\mathbf{v}} f(\mathbf{v_{S_g}})]^T$ with respect to $\mathbf{v}$.
14: Compute $\mathcal{C}_f$ and its eigendecomposition: $\frac{1}{S_f} \sum_{r=0}^{S_f} \nabla_{\mathbf{v}} f(\mathbf{v})\ \nabla_{\mathbf{v}} f(\mathbf{v}_q)^T = \mathbf{W}^f \boldsymbol{\Lambda}^f \mathbf{W}^{fT}$
15: Partition eigenspace of $\mathcal{C}_f$ as in Eq. (7.5), which is composed having maximum separation between last element of $\boldsymbol{\Lambda}_1^f$ and first element of $\boldsymbol{\Lambda}_2^f$ (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., $\mathcal{U} := \{u_k^l \leq u_k \leq u_k^u,\ \forall k \in \{1, 2, \ldots P\}\}$).
17: Sample $\mathcal{U}$ to create set $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_{S_r}]^T$, where S_r sample size of the training dataset for building surrogate model.
18: Evaluate QoI for $\mathbf{U}$, $\mathbf{G} = [g(\mathbf{u}_1), g(\mathbf{u}_2), \ldots, g(\mathbf{u}_{S_r})]^T$
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. (7.41) 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. (7.42).

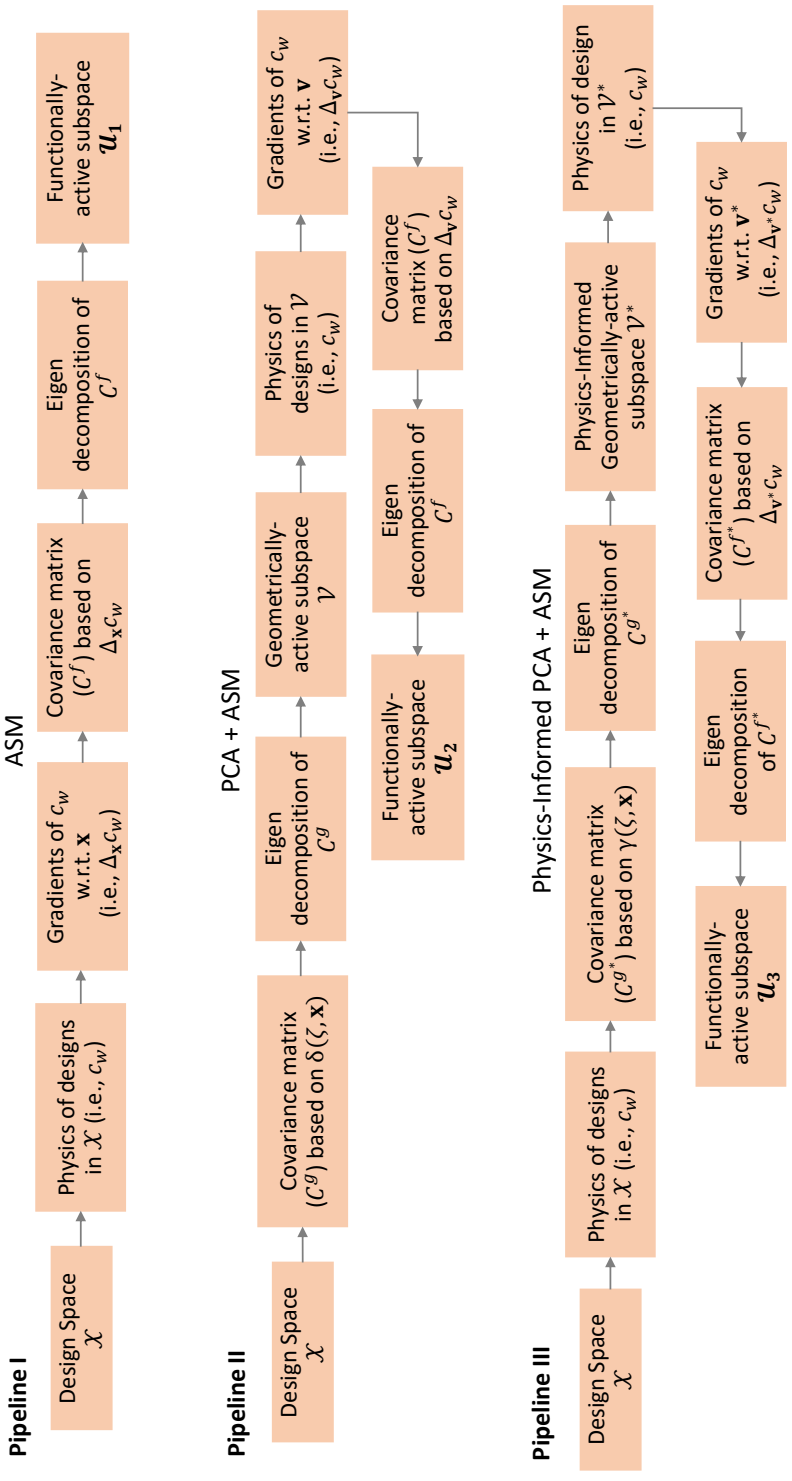


Figure 7.3: Sequential workflow of testing pipelines used for the experimentation and validation of the proposed approach.

the discussion on surrogate modelling and shape optimisation, we explain the results of the FAS formed with original design parameters and compare them with those formulated with two-step feature extraction.

7.3.1 Dimensionality reduction with geometric features

Extraction of geometric features to form $\mathcal{V}$ or $\mathcal{V}^*$ is commenced with the eigendecomposition of $\mathcal{C}^g$ or $\mathcal{C}^{g^*}$, which accumulates the geometric variability of designs into eigenvalues and eigenvectors. The former identifies the original geometric variability of the full design space in terms of geometry or physics. In contrast, the latter contains the extracted feature for the latent variable and defines the shape deformation modes of the 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 [154]. 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 same settings as described in §5.3.2. Fig. 7.4 (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 Fig. 7.4, it can be seen that for the present test case under the specified setting that the first 15 and 11 eigenvectors, $\mathbf{W}_1^g = \{\mathbf{w}_i^g, i = 1, 2, \ldots, 15\}$ and $\mathbf{W}_1^{g^*} = \{\mathbf{w}_i^{g^*}, i = 1, 2, \ldots, 15\}$, obtained from the eigendecomposition of $\mathcal{C}^g$ and $\mathcal{C}^{g^*}$, retains 95.18% and 95.34% of geometric variance, respectively. This results in approximately 44% ($N = 15$) and 59% ($N^* = 11$) reduction of the original design space's dimensionality to create 15-

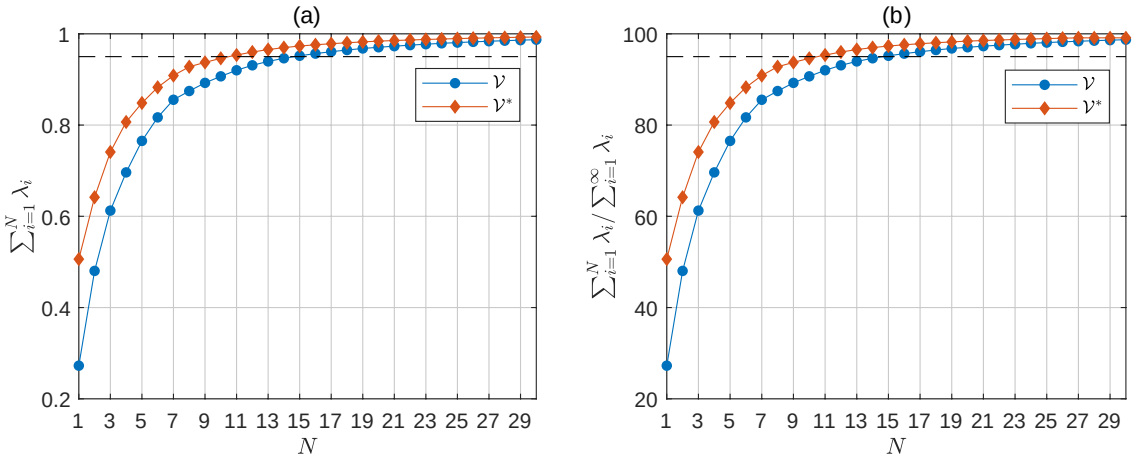


Figure 7.4: Plot showing (a) absolute and (b) percentage of the geometric variance retained by geometrically-active ($\mathcal{V}$) and physics-informed geometrically active ($\mathcal{V}^*$) subspaces of dimension N. Horizontal lines on plot (a) and (b) indicates the threshold for 95% of the geometric variance.

and 11-dimensional $\mathcal{V}$ and $\mathcal{V}^*$, respectively. This also quantifies that only $N = 15$ or $N^* = 11$ latent directions govern or accumulate all the geometric variations in $\mathcal{X}$. It is also noteworthy that in the presence of c_w the higher geometric variance is achieved and its first eigenvector captures 50.61% of the variance, which is approximate twice the one captured by the first eigenvector of $\mathcal{C}^g$. Moreover, up to $N, N^* = 20$, eigenvectors obtained from $\mathcal{C}^{g^*}$ contain a higher cumulative sum of eigenvalues compared to ones obtained from $\mathcal{C}^g$; however, it tends to be the same after 20^{th} eigenvector.

Figures 7.5 (a) and (b) show the 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) [160], 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. 7.6. 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, it is second ($\mathbf{w}_2^{g^*}$) and third ($\mathbf{w}_3^{g^*}$) eigenvector shares a 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 quantify that the eigenvectors for $\mathcal{V}^*$ capture more of 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 compared to physics, into the dataset and the designs' parametric discretisation. Although this would require the validation of integrals' accuracy and decision on the satiable order of integrals to achieve a concrete descriptor, this seems to be an interesting study, and the authors plan to work in the future.

7.3.2 Dimensionality reduction with functional features

As explained previously, the extraction of functional features of a 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.

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, one could use finite-difference

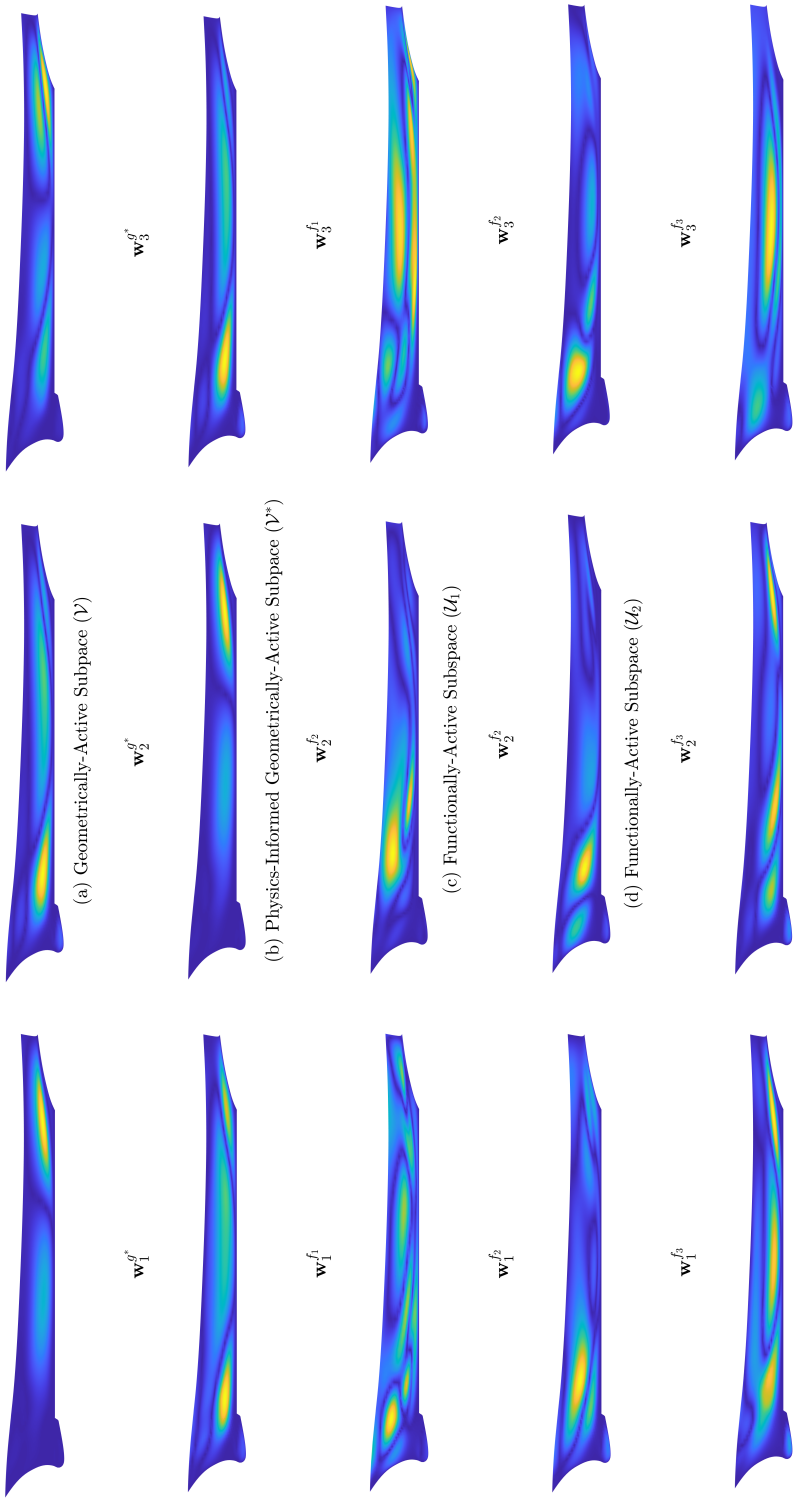


Figure 7.5: Shape deformation modes are corresponding to (a) first, (b) second and (c) third eigenvectors obtained for geometric and functional variance retained by geometrically-active ($\mathcal{V}$), physics-informed geometrically active ($\mathcal{V}^*$), and functionally-active subspaces; $\mathcal{U}_1$, $\mathcal{U}_2$ and $\mathcal{U}_3$. The colours indicate the magnitude of normal surface displacement.

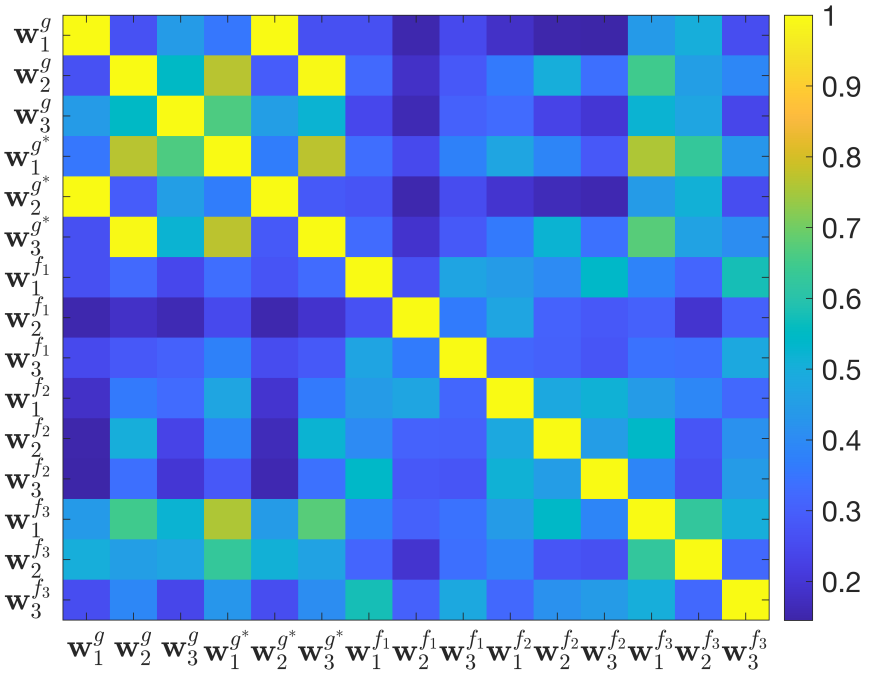


Figure 7.6: Modal assurance criterion plot to compare the first three eigenvectors of the geometrically-active, physics-informed geometrically-active and functionally-active subspaces.

or evaluate gradients using adjoint solvers if the based 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, primarily evaluated from adjoint solvers, are qualitatively correct. They have the right relative scaling and right sign. However, quantitatively, in terms of magnitude, they are not reliable [97, 98], at least when they are used for active subspace.

Therefore, once could approximate these gradients with a local surrogate model, as we are interested, in active subspace, in the local behaviour of the problem when approximating gradients. Therefore, inventors of ASM proposed an algorithm for building *local model-based* gradients, in which a local linear model is fitted with a subset of predictions from the training dataset [79].

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. [79] recommends using least squares to fit a local MLR model, but from our experiments, we found a local GPR model with a linear basis fits the subset with better accuracy and is thus utilised in the present study. Fig. 7.7 (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 from these models for the baseline hull design are shown in Fig. 7.7 (c). Setting an appropriate value for ℓ also requires some attention, which we will discuss in the next paragraph. The R^2 and MSE for the GPR model are 0.97152 and 0.16709, and for the MLR model, these values are 0.66291 and 0.58155, respectively. From Fig. 7.7 (a) and (b), it is clear that GPR well approximates the problem. Therefore, gradients evaluated from this model are of better quality. Analysing Fig. 7.7 (c), it can be seen for most of the parameters, gradients evaluated from MLR are lower in magnitude compared to the ones evaluated from GPR. Moreover, for parameters 6, 13 and 14, gradients from MLR

also have opposite signs compared to gradients evaluated from GPR. Therefore, we used model-based gradients approximated with local GPR for functional features in all three test cases.

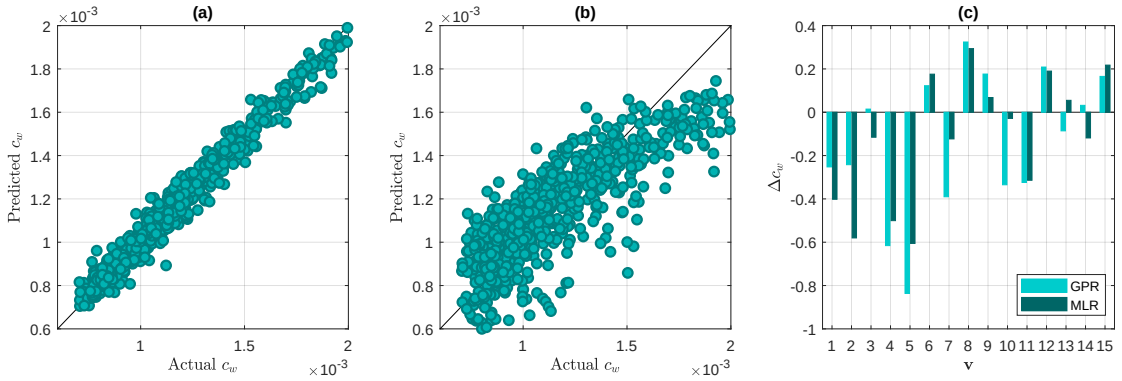


Figure 7.7: Sufficient summary plot between actual and predicted wave resistance coefficients (c_w) evaluated using Gaussian Process Regression (GPR) and Multiple Linear Regression (MLR). (b) Comparison of model-based gradients obtained using GPR and MLR models.

As mentioned in the previous paragraph, the value of ℓ also has 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, [79] 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, 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 ran 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 Fig. 7.8. From this Figure, it can be observed that from $\ell = 50$ to $\ell = 200$ gradients of c_w, Δc_w, fluctuate 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. After certain ℓ values, the model-based gradients were constant throughout the sampled space or subspace for three test pipelines.

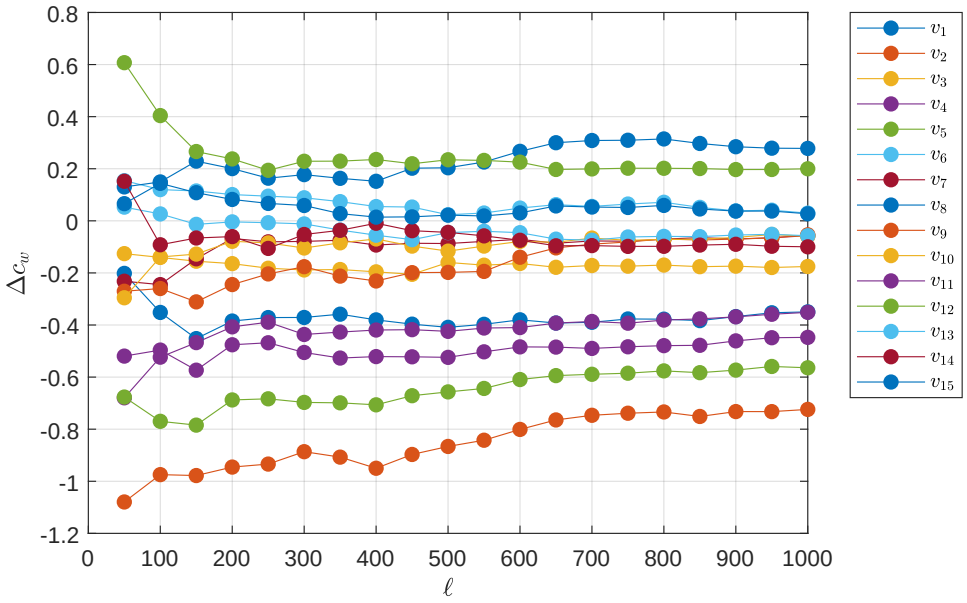


Figure 7.8: Plot for gradients of wave resistance coefficient (Δc_w) versus the size of the training dataset (ℓ) used to construct the GPR model for the evaluation of model-based gradients of the baseline test model.

Extracting features

As explained earlier for FAS, 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 separation between their eigenvalues. These P features create latent variables for a P-dimensional FAS.

Fig. 7.9 (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 Fig. 7.3. The plot in Fig. 7.9 (a) shows the prominent separation between the first and second eigenvalues

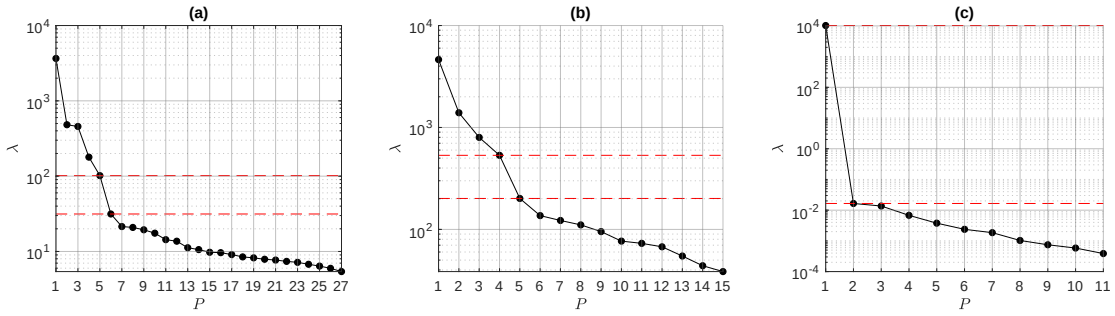


Figure 7.9: Eigenvalue decay versus the dimensionality (P) of functionally-active subspace representation of (a) original design space, (b) geometrically-active and (c) physics-informed geometrically-active subspaces.

and between the 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 the second prominent separation, the first five eigenvectors $\mathbf{W}_1^{f_1} = \{\mathbf{w}_i^{f_1}, i = 1, 2, \ldots, 5\}$ 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. 7.9 (b) shows the eigenvalue plot for the second test pipeline, which is the two-step feature-to-feature learning. In this case, the functional features of $\mathcal{V}$ are extracted, further reducing its dimensionality. From Fig. 7.9 (b), it can be seen that the largest separation occurs between the fourth and five eigenvalues. This results in a $P_2 = 4$-dimensional $\mathcal{U}_2$, span by the first four eigenvectors $\mathbf{W}_1^{f_2} = \{\mathbf{w}_i^{f_2}, i = 1, 2, \ldots, 4\}$, resulting in 85% reduction in the dimensionality of $\mathcal{X}$. It can be seen that, in this case, compared to $\mathcal{U}_1$, there is no notable reduction achieved with two-step feature extraction. However, as we will see in subsequent sections, in terms of surrogate modelling and optimisation, $\mathcal{U}_2$ captures better geometric and functional variability.

As explained previously, in the third test pipeline, the first geometric decomposition happens not only in terms of geometry but also in terms 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. 7.9 (c). These results show a significant separation

between the first and second eigenvalue, resulting in one-dimensional $\mathcal{U}_3$ (i.e., $P_3 = 1$) with the first eigenvector $\mathbf{W}_1^{f_3} = \{\mathbf{w}_1^{f_3}\}$. The significant separation between the first and second eigenvalue happens because the basis of the physics-informed geometric representation of the 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 $\mathbf{w}_1^{f_3}$.

Fig. 7.5 (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 Fig. 7.6. The 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 the first three eigenvectors of $\mathbf{W}_1^{f_1}$ and $\mathbf{W}_1^{f_2}$ do not share any similarity as their MAC values are 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 a significant likeness.

Sensitivity analysis

The functional features can give helpful insight into 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's total sensitivity indices and derivative-based global sensitivity measures [80]. The eigenvectors identify the most important direction $\mathcal{X}$, 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:

$$\phi_i = \phi_i(P) = \sum_{j=1}^{P} \lambda_j w_{i,j}, \quad \text{where, } i = 1, 2, \dots P \tag{7.43}$$

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 the following:

$$\phi_i \mapsto \frac{\phi_i}{\sqrt{\sum_{i=1}^{P} \phi_i^2}}. \tag{7.44}$$

For further details, interested readers should refer to ref. [80]. Fig. 7.10 (a), (b) and (c) shows the activity score for $\mathbf{x}$, $\mathbf{v}$ and $\mathbf{v}^*$, respectively. From Fig. 7.10 (a), it can be observed that the most significant parameter for $\mathcal{X}$ is x_{14} followed by x_8, x_{15} and x_4, where x_1 is the least significant parameter. In case of $\mathcal{V}$ (Fig. 7.10 (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 (Fig. 7.10 (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.

7.3.3 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 affected by the outliers in the training dataset, which can result in inaccurate or mediocre prediction of c_w. Therefore, a univariate method based on the quartiles is used to detect and remove outliers [161], which identifies an element of the dataset as an outlier if its value is more than 1.5 interquartile

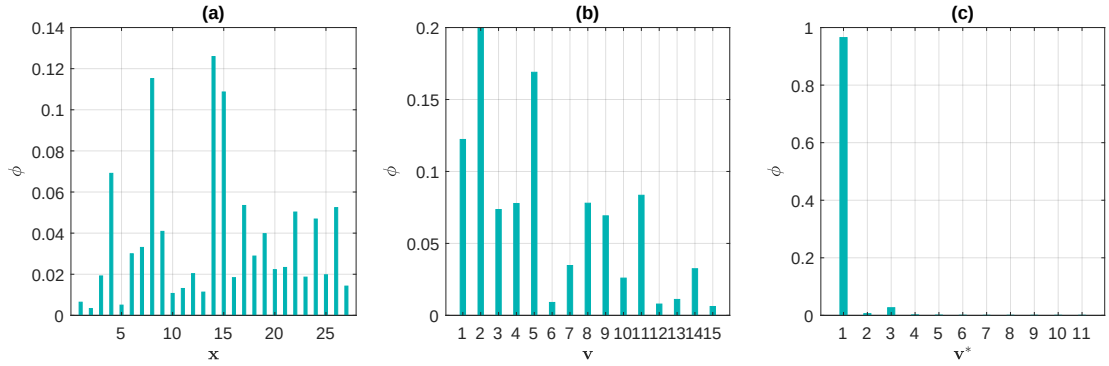


Figure 7.10: Plot showing activity/sensitivity scores (ϕ) for the (a) design parameters, $\mathbf{x}$, (b) geometrically-active latent variables, $\mathbf{v}$ and (c) physics-informed geometrically-active latent variables, $\mathbf{v}^*$.

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 different kernel functions will be discussed in the subsequent sections. For training validation, 10-fold cross-validation is implemented. Table 7.1 shows the results of training-MSE, which is evaluated between actual and predicted values of c_w in the training dataset. 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 7.1,

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 a value close to one shows good prediction. Thus a model with higher R^2 should be selected. From the results of Table 7.1, one can see that g_3, which is constructed from $\mathcal{U}_3$, has the lowest training error followed the by g_2 and g_1. These results show that feature-to-feature dimensionality reduction does

Table 7.1: 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

help to elevate the surrogate modelling accuracy. Fig. 7.11 (a) shows the plot of MSE versus dimensionality of g_1, g_2 and g_3. It is noteworthy that for all three models, the MSE decreases as their dimensionality increases. However, for g_3, this change is negligible (see Fig. 7.11 (b)) as even with one parameter, g_3 has significantly higher accuracy compared to the 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 align with the results of the eigen-decomposition 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 by increasing the dimensionality. However, this will increase the computational cost of 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 than 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 terms 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.

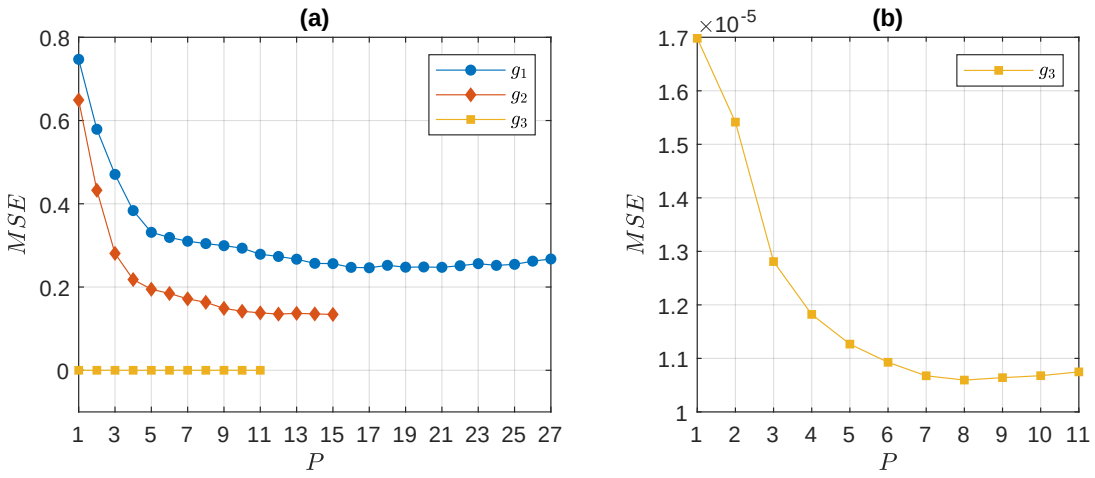


Figure 7.11: Plot showing Mean Square Error (MSE) versus dimensionality (P) of surrogate models g_1, g_2 and g_3 trained, respectively, with designs sampled from $\mathcal{U}_1$, $\mathcal{U}_2$ and $\mathcal{U}_3$. (b) A magnified version of the plot between MSE and dimensionality of g_3.

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), and 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. 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 7.12, which show an interesting behaviour that the linear techniques, MLR

and Linear SVM, cannot map well the nonlinear behaviour of the problem; thus, compared 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 outperformance other variants of SVM. Subsequently, it is interesting to note that compared 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 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 training-MSE and R^2.

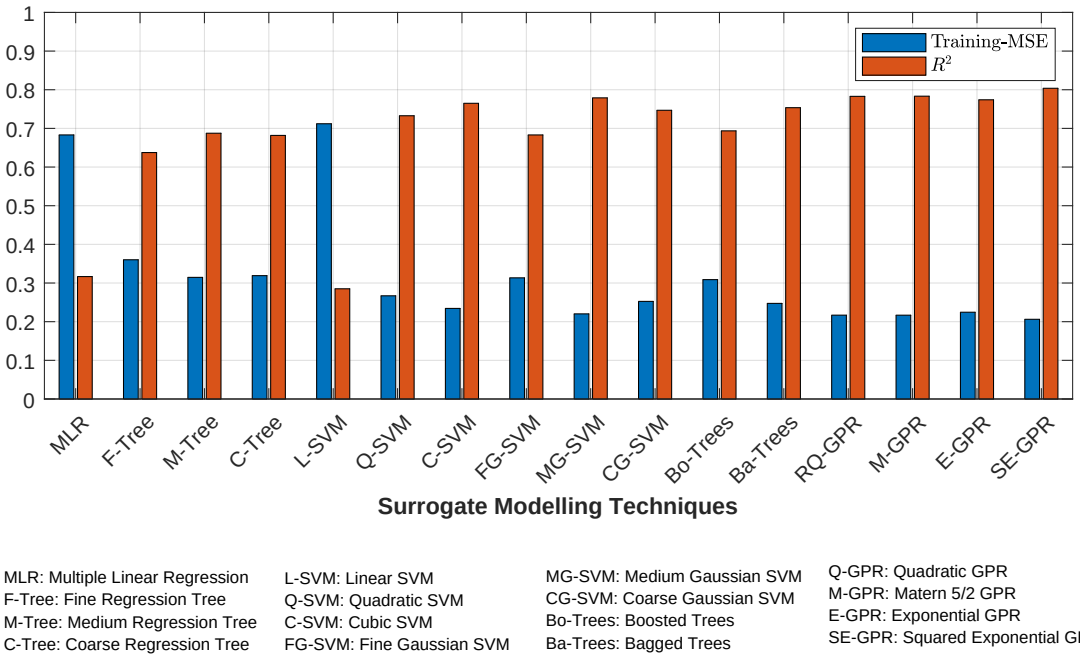


Figure 7.12: Comparison of different techniques used to construct g_2.

7.3.4 Optimisation

To optimise the baseline hull, we utilised the Jaya Algorithm (JA) [109], 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 requiring a set of randomly sampled initial solutions to optimise. Unlike most population-based meta-heuristic optimisation techniques, JA does not require tuning 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 7.2.6 to reduce the model-scale calm-water wave resistance coefficient, c_w, of the baseline ship model. As JA is a stochastic meta-heuristic technique that may provide different results in each run, 100 different optimisation runs were performed. In each run, a total of 1500 iterations were conducted. Fig. 7.13 (a) shows the average values of c_w in 100 runs, and Fig. 7.13 (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 7.2.

Table 7.2: 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

It is noteworthy that $\mathcal{U}_2$ gives the most optimal design followed by $\mathcal{U}_1$ and $\mathcal{U}_3$. Interestingly, despite showing higher accuracy both in terms 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 overlearnt the variability of c_w. Therefore, the optimiser cannot explore diverse designs for optimum global compared to $\mathcal{U}_1$ and $\mathcal{U}_2$. Fig. 7.14 (a), (b) and (c) shows the shape difference be-

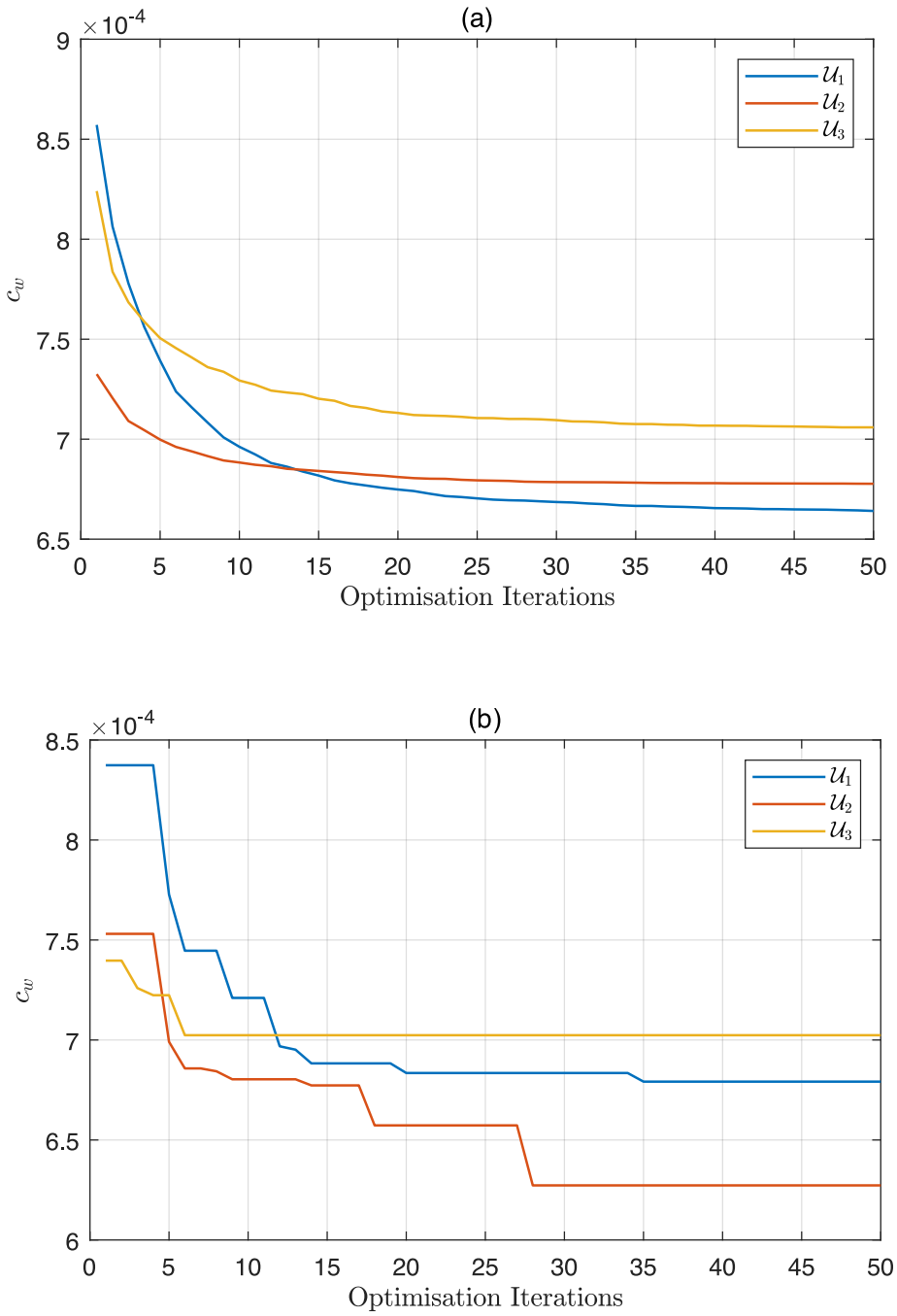


Figure 7.13: Plots for the objective function (c_w) versus a number of optimisation iterations performed in functionally-active subspaces of Fig. 7.3. (a) Average c_w over 100 optimisation runs and (b) c_w in a single optimisation run over the first 50 iterations.

tween the baseline design and optimised designs obtained, respectively, from $\mathcal{U}_1, \mathcal{U}_2$ and $\mathcal{U}_3$. Table 7.3 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 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 7.3 that these parameters are close to each other with mirror variations. Although these variations can be reduced with design constraints during optimisation, they are not critical, at least in the present work context.

Compared 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 Fig. 7.13 (b), it is also noteworthy that optimisation performed in $\mathcal{U}_2$ converges 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 7^{th} 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. Five-dimensional g_3 was developed to evaluate designs in the five-dimensional version of $\mathcal{U}_3$. The optimal design obtained at the end of optimisation has $c_w = 0.00070373$, which shows slight improvement compared to the one obtained from one-dimensional $\mathcal{U}_3$. Moreover, this design still has lower performance than 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.

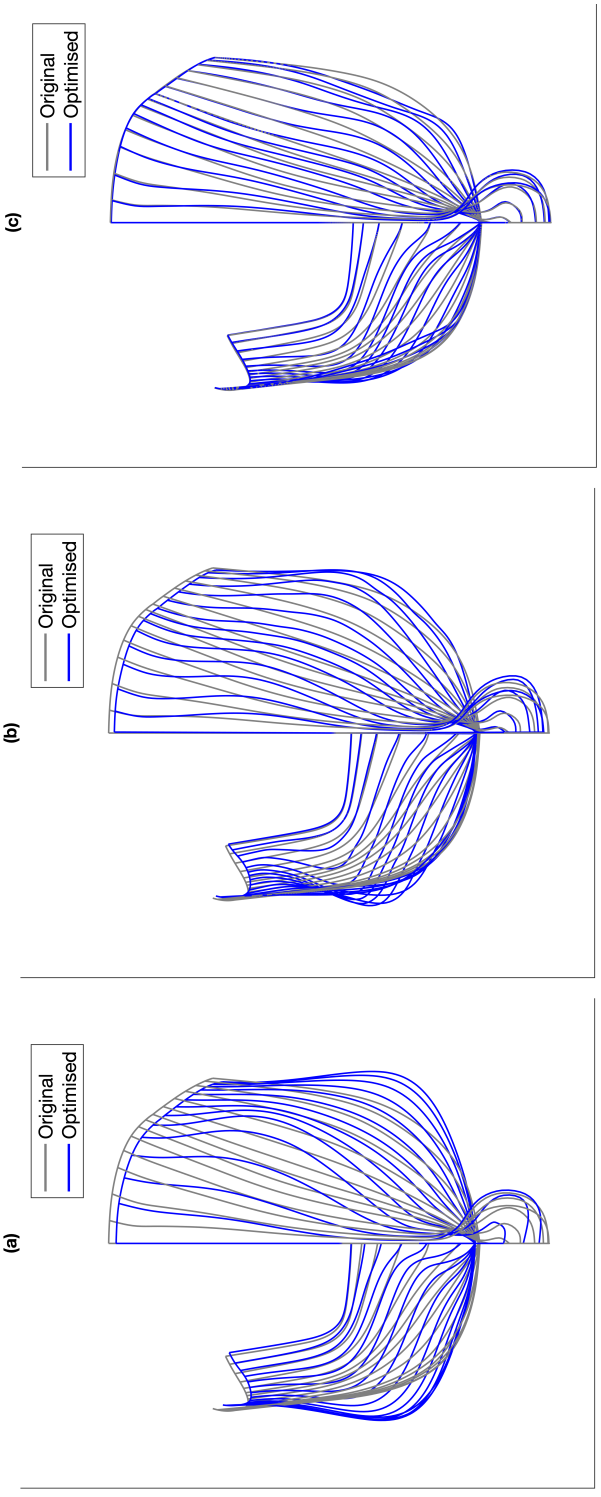


Figure 7.14: Comparison between the baseline design and optimised designs obtained, respectively, from functionally-active subspaces (a) $\mathcal{U}_1$, (b) $\mathcal{U}_2$ and (c) $\mathcal{U}_3$.

Table 7.3: 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	$\mathcal{U}_1$	$\mathcal{U}_2$	$\mathcal{U}_3$
$\boldsymbol{T}$	6.17000	6.17000	6.17000	6.17000	$\boldsymbol{LCF}$	77.8780	77.4501	79.0769	78.6429
$\boldsymbol{\nabla}$	2974.23	3129.98	3193.92	3048.54	$\boldsymbol{A_w}$	2095.02	2208.29	2292.37	2153.80
$\boldsymbol{S}$	8419.32	8425.02	8432.30	8429.33	$\boldsymbol{A_m}$	96.0803	106.271	100.510	94.2810
$\boldsymbol{Lwl}$	142.050	142.050	142.050	142.050	$\boldsymbol{C_p}$	0.61688	0.55810	0.59059	0.62940
$\boldsymbol{Bwl}$	19.0810	21.0200	20.1320	19.8810	$\boldsymbol{C_m}$	0.81610	0.81940	0.80917	0.76864
$\boldsymbol{B}$	20.5390	20.2057	19.8714	20.4606	$\boldsymbol{C_w}$	0.77294	0.73957	0.80160	0.76260
$\boldsymbol{LCB}$	71.8693	71.8677	72.4424	71.5878	$\boldsymbol{C_b}$	0.50344	0.45731	0.47789	0.48378
$\boldsymbol{VCB}$	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.

The efficiency of one-dimensional g_1 and g_2 differs 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. (7.45).

$$
\begin{aligned}
\text{given} \quad & \mathbf{u}' = \mathbf{W}_1^{f'^T}\mathbf{v} \in \mathcal{U}',\ \mathbf{W}_1^{f'} \in \mathbb{R}^{N\times P'} \\
& \mathbf{u}'' = \mathbf{W}_1^{f''^T}\mathbf{v} \in \mathcal{U}'',\ \mathbf{W}_1^{f''} \in \mathbb{R}^{N\times P''} \text{ where } P' < P'' \\
& \mathbf{W}_1^{f*'} \in \mathbb{R}^{P''\times P'} \\
& c_w \approx g'\left(\mathbf{u}''\right) \\
\min_{\mathbf{u}'} \quad & g'(\mathbf{W}_1^{f*'}\mathbf{u}') \\
\text{subject to} \quad & \mathbf{u}^{l'} \leq \mathbf{u}' \leq \mathbf{u}^{u'} \\
& \mathbf{u}^{l''} \leq \mathbf{W}_1^{f*'}\mathbf{u}' \leq \mathbf{u}^{u''} \\
\text{yield} \quad & \mathbf{u}'_{optimal}
\end{aligned}
\tag{7.45}
$$

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. (7.45) defines the viable search space for exploration of $\mathbf{u}'_{optimal}$. Whereas the second constraint in Eq. (7.45) ensures that the design evaluated with g' lies within its domain on which this model was initially trained. The evaluation of design's c_w, which lie outside $\mathcal{U}''$, will result in its false/inaccurate estimation, resulting in guiding the optimisation towards local optima.

Table 7.4: 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 the 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 7.4 summarises all the optimisation tests performed under different dimensionality of subspaces and surrogate models. These results again demonstrate that the dimensionality reduction performed with feature-to-feature learning encodes the lower-dimensional latent subspace well compared to the one performed with signal-step feature extraction.

7.4 Conclusions & future works

This chapter proposed a dimensionality-reduction method to reduce the computational cost and increase the efficiency of SDO. The proposed approach commences with feature extraction to learn a lower-dimensional latent space in which the 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 was extracted to reduce its dimensionality. This two-stage feature-to-feature learning creates a functionally-active subspace while re-

taining the geometric and functional variability of the original design space. The geometric features help optimisation to explore diverse designs in a lower dimension, and functional features facilitate efficient surrogate model training. We tested the proposed approach on a 27-dimensional design space to optimise the 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 improves SDO's computational efficiency.

As a future work, authors are keen to work on the development of a Psycho-Physical metric [8] to visually access the geometric variance retained by a certain subspace, which might facilitate designers to explore subspaces in a generative design paradigm effectively.

Chapter 8

ShipGAN: Deep convolutional generative model for parametric ship design

8.1 Introduction

> "*What I cannot create, I do not understand.*"
>
> **Richard Feynman**

Recently, the increasing pervasiveness of machine learning in engineering design, mainly in the form of the emerging scientific machine learning (SciML), has, on several occasions, taken off the vast computational burden from traditional solvers by constructing efficient low or even high-fidelity surrogate models that almost instantaneously predict performance, thereby accelerating the entire simulation-driven design (SDD) pipeline. Although the efforts of integrating SciML in ship design are increasing, the pace is relatively slow compared to other engineering fields.

Furthermore, there are few efforts to introduce these tools at the preliminary ship de-

sign stage, where naval architects and/or involved designers typically identify designs from existing databases while attempting to match new requirements. Afterwards, they may construct a parametric model using a suitable ship-hull surface representation, typically comprising NURBS surface patches or simpler panel meshes. This usually results in a narrow design space permitting only slight variations of a baseline design [162]. Designers also get inspiration from existing designs while using their features and components to create a small set of potential alternatives. However, embedding these features is a complicated task and constructing a new parametric description for the unique shape using existing strategies is highly expertise-driven and time-intensive. Although this approach has served the community well, especially when considering well-established ship types, we may need to leap forward toward radical design ideas when uncommon requirements demand the exploration of a richer design space or if new business models require revolutionising and redesigning existing ship types. This obviously benefits novel design tasks, e.g., *special purpose vessels*, but it can also offer a competitive advantage for traditional players in the industry.

There have been substantial efforts in computer-aided ship design for building robust parametric tools, but they can only handle a specific hull type; some relevant examples of such tools are presented in [1, 2, 4, 54, 163, 164]. Despite their efficiency in creating valid and smooth ship-hull geometries, they cannot be readily used to generate instances of ship types that deviate significantly from their target ship types. For example, in Fig. 8.1, the parametric construction proposed by [1], and later explicitly adapted for container ship hulls by [2], is depicted. Such parameterisation cannot be directly or easily mapped to an entirely different ship-hull type, such as the DTMB naval ship shown in the same figure. Although some generic approaches, like FFD (free-form deformation) [61], may be applicable to some extent, they either use a rather crude low-fidelity and featureless representation or require significant effort and experimentation for adaptation into new designs. For example, FFD-based parameterisations are not truly feature-driven [18], which deprives designers of the commonly needed feature-modelling capabilities and local control

for designs such as bulbous bows or other features of local nature.

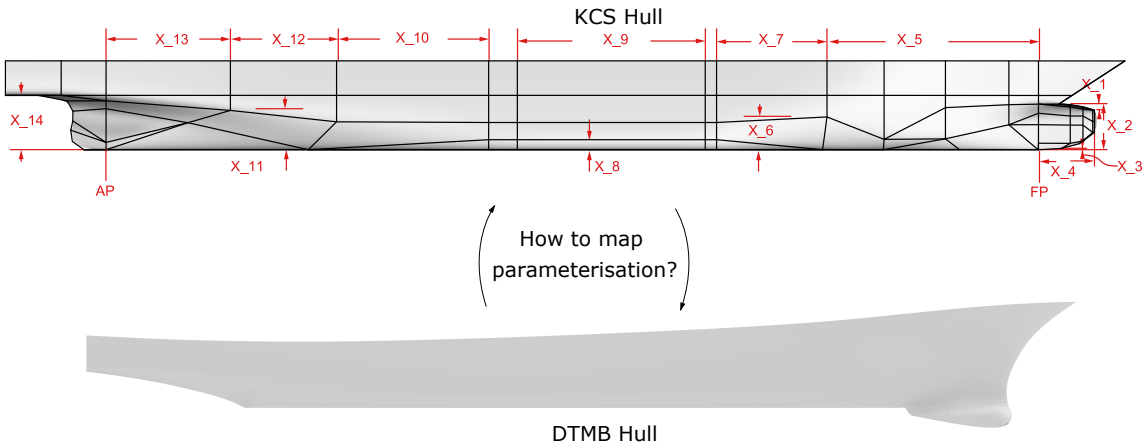


Figure 8.1: The Parameterisation proposed by [1,2] for container ship hulls. Is it applicable to a naval ship design such as the DTMB hull?

In this work, we aim to tackle the above-mentioned challenges in a typical parametric hull design by proposing a generic parametric modeller, ShipGAN. The new model can handle various ship hull types and transform one type into a completely different one, as shown in Fig. 8.2. The proposed modeller uses deep generative models, specifically deep convolutional generative adversarial networks (GANs) [165, 166], with a new architecture and loss function suitable for the problem at hand. These generative models were initially proven to be promising for generating entirely novel images from given datasets and recently have been exploited for engineering design problems, i.e., aerodynamic design and optimisation [32, 166]. If appropriately trained, they can efficiently learn latent representations, which can then be used as design parameters to construct diverse design spaces for shape optimisation. However, the capacity of these approaches has not been explored in ship design.

Despite their proven efficiency in design, these models have their limitations. Since they were initially developed for 2D datasets, e.g., processing of images, their application in 3D design requires *suitable geometric representations* to extract meaningful features. An inappropriate training of such models can therefore result in many invalid shapes. More

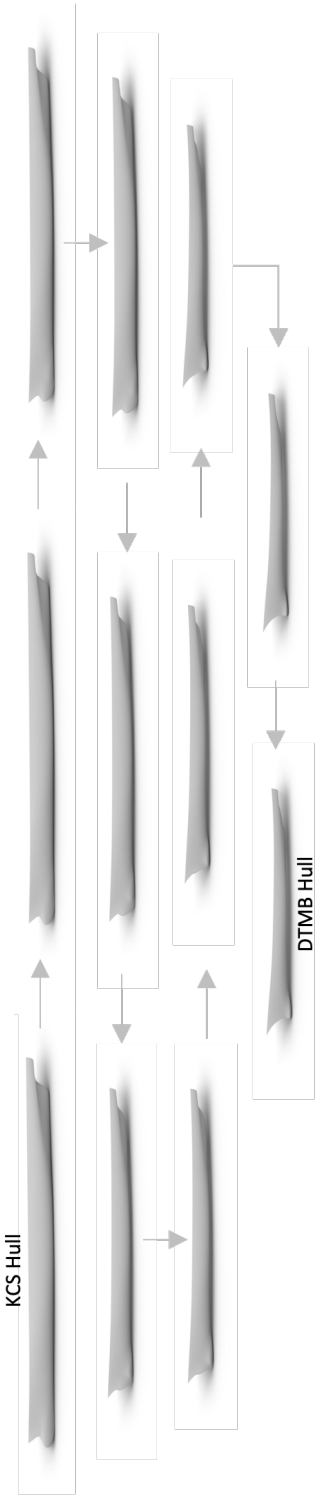


Figure 8.2: Transformation of KCS hull into DTMB hull using our ShipGAN parametric modeller.

importantly, if the dataset is composed of various design sub-classes, they also tend to lose in generalisability [96].

We, therefore, propose a modified architecture and a loss function to overcome the drawbacks inherited from GAN. To commence the training of ShipGAN, we first developed a technique to transform different types of ship hulls into a common geometric representation. Furthermore, we constructed a shape-signature tensor (SST) using appropriately encoded designs and their geometric moments (GMs) [132]. Therefore, the so-constructed SST augments and enriches the geometric information related to designs given to the ShipGAN model during training by infusing the moment-related physics associated with ship hulls. In this way, SST acts as a unique descriptor of each dataset design instance that enables the extraction of meaningful features which are not only geometry-driven but also physics-integrated to provide rich and physically-valid design alternatives. We use a deep convolutional architecture [165] for the model to capture sparsity in the training dataset, along with a space-filling term [99] in the loss function to enhance diversity.

To the best of the authors' knowledge, this is the first attempt to construct a generic parametric modeller in the field of parametric computer-aided ship design. In accordance with the aim of this work, we report the following main contributions:

1. Development of a large shape dataset containing more than fifty thousand design variations of several existing classes of ships; no such extensive dataset is publicly available.

2. Development of an intuitive approach to convert all ship designs into a common geometric representation. This technique also ensures a smooth NURBS-based reconstruction of designs resulting from the model.

3. The combination of a geometry descriptor with its relevant geometric moments enables capturing of global and local geometric features with physics-informed elements in the latent space, which in turn allows the generation of designs that are both geometrically valid and physically plausible.

4. Introduction of a space-filling term to the loss function, which enables the model to cover the entire spectrum of the training dataset, thereby enhancing diversity and potentially avoiding mode collapse issues [167].

5. We finally conducted extensive comparative studies that showed ShipGAN outperforms typical GANs in design diversity, quality and validity. We also demonstrated the usability and generic capabilities of ShipGAN in the ship design cycle via large-scale ship-hull optimisation experimentation.

8.2 Background on generative adversarial networks

This section provides a brief but essential introduction to typical GANs, i.e., Vanilla GAN, and their applications in engineering design and optimisation. A typical GAN model consists of two neural networks, generator G and discriminator D, which are trained simultaneously to enhance the capability of G to map from a latent space to the data distribution of interest and thus aim to generate new designs which could have been part of the real designs dataset. In contrast, D tries to classify designs, i.e., to distinguish between real (designs in the training dataset) and generated designs, also referred to as fake designs. Networks G and D are trained simultaneously to reach a Nash equilibrium with the following minimax loss function:

$$\min_G \max_D \mathcal{L}_{adv}(D,G) = E_{\mathbf{x}\sim p_{data}(\mathbf{x})}[\log(D(\mathbf{x}))] + E_{\mathbf{z}\sim p_{\mathbf{z}}(\mathbf{z})}[\log(1-D(G(\mathbf{z})))] \tag{8.1}$$

where $\mathbf{x}$ represents designs in the training dataset and $\mathbf{z}$ denotes the latent tensors randomly sampled from a given distribution $p_{\mathbf{z}}$. The training of the GAN is typically seen as a game or competition between G and D, thus referred to as adversarial training, which facilitates learning the data distribution $p_{data}(\mathbf{x})$ of real designs $\mathbf{x}$. During training, the performance of D is maximised so that it can accurately distinguish $\mathbf{x}$ from the synthetic

designs, $G(\mathbf{z})$, sampled from $p_{\mathbf{z}}$. During this training, G minimises $\log(1 - D(G(\mathbf{z})))$ to learn to produce designs that the discriminator will classify as real designs, i.e., designs resulting from the generator will tend to be similar to real designs.

The adversarial training commences with mini-batches of samples from $p_{\mathbf{z}}$, and G tries to produce realistic designs based on these samples. Then, D is trained to identify whether the presented designs are real (i.e., from the training dataset) or fake (i.e., from the generator). During this process, both networks adjust/optimise their parameters to outperform their opponent, i.e., as D improves its classification ability, G also enhances its ability to create data that fools D. This process continues until convergence is achieved. This way, G of the trained GAN model can generate new designs with sufficient diversity within the prior distribution.

Both G and D can be nonlinear mapping functions, such as a conventional neural network (NN) or a convolutional NN (CNN). In our case, we use CNN as they have been proven more effective in capturing sparse features. D and G with CNN-like architecture are often referred to as Deep Convolutional GAN (DCGAN) [165, 166].

8.2.1 GANs in engineering design

GANs and their variations have been used for various tasks; however, in this work, we focus on their application in engineering design. Recent applications of GANs in the context of engineering design have appeared in topology optimisation [168, 169], design and optimisation of aerofoils and wings [32], design of metamaterials [170] and synthesis of design creativity in bicycle design [171].

Chen et al. [32] proposed a Bézier-GAN model for airfoil design and optimisation. To achieve a high representation capacity (i.e., design variation) and compactness (i.e., design validity), Bézier-GAN uses a Bézier curve layer right after the generator, which fits a Bézier curve to data sampled from the employed distribution. Later, Chen et al. in [172] proposed a Bézier-GAN variation based on conditional GANs, called CBGAN, to mainly tackle the inversion ambiguity in the inverse design of aerofoils. A performance-conditioned

diverse GAN (PcDGAN) was proposed by Nobari et al. [173], which uses a new self-reinforcing score called Lambert Log Exponential Transition Score (LLETS) for improved conditioning. Chen and Ahmed proposed a performance-augmented diverse generative adversarial network (PaDGAN) [96] and its multi-objective extension MO-PaDGAn [174] to ensure that the trained generator remains applicable, with good-performing designs, outside the training dataset domain. To achieve this objective, PaDGAN uses a new loss function based on determinantal point processes (DPPs), which tries to maximise the spread of designs based on their geometric similarity and performance. However, PaDGAN requires the evaluation of performance and its gradients, which is commonly computationally expensive to evaluate. This problem is tackled in the present work by using geometric moments (GMs) as a physics-informed performance descriptor instead of directly employing performance evaluations.

To detect geometric abnormality of generated aerofoils or wings, Li et al., [166] trained a DCGAN with a discriminative model based on convolutional neural networks, which detects invalid designs without the need of a separate and expensive computational evaluation. Chen and Fuge [175] proposed a hierarchical GAN model to allow the synthesis of designs with interpart dependencies. Nobari et al. [176] trained a conditional GAN model to enforce the generator to create designs within a specific performance range and tested their network in the generation of 3D shapes corresponding to aeroplanes. A CreativeGAN model was proposed by Nobari et al. [171] to ensure the generation of novel design alternatives. To enhance novelty, CreativeGAN used the K-nearest neighbour (KNN) approach to detect novel features of designs and use these features to train StyleGAN model [177], which is capable of generating designs with the detected novel features.

8.3 ShipGAN

In this section, we provide an in-depth presentation of the ShipGAN model considerations and its architecture, schematically depicted in Fig. 8.3. The generator and discriminator of

the proposed model have a deep convolutional architecture to better capture the sparsity in the data. ShipGAN uses space-filling [99] to evenly capture the diversity present within the training dataset and SST to inject the notion of physics in the latent features during training.

Let $\mathcal{G}$ be a geometric object representing a baseline design (e.g., a parent hull) in an ambient space $\mathcal{A} \subseteq \mathbb{R}^3$. We also assume that $\boldsymbol{P}(\mathcal{G})$ is a vector function $\mathbb{R}^{\bar{n}} \to \mathcal{A}$ that provides the geometry of the object $\mathcal{G}$, $\mathbf{x} = \mathcal{P}(\mathcal{G})$. Along with $\mathbf{x}$, there is a lumped geometric moment vector, $\boldsymbol{M}(\mathcal{G}) \in \mathbb{R}^{n_M}$. Now combining the geometry and its moments results in a unique SST,

$$\text{SST} = \big(\boldsymbol{P}(\mathcal{G}), \boldsymbol{M}(\mathcal{G})\big)\,, \tag{8.2}$$

encompassing high-level information about the baseline design. Therefore, the training dataset comprises the set of shape designs, $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \ldots, \mathbf{x}_n\}$ and the corresponding GMs of each design.

8.3.1 Shape dataset

SciML for engineering design problems suffers mostly from inappropriate and/or insufficient amounts of data. This is especially challenging if labels, typically performance parameters, are evaluated by time-consuming high-fidelity solvers. However, generative models are generally unsupervised and do not require labels; nevertheless, a sufficiently diverse dataset with novel design alternatives is required to acquire a trained model with good generalisability. In the context of engineering design, application of such models has so far appeared in automotive [178] and aerofoil [32] design, since relevant datasets such as shapeNet[1] and UIUC airfoil coordinates database[2], containing several thousand designs, are publicly available. To the best of the authors' knowledge, no equivalent, diverse and publicly available dataset of ship-hull designs exist.

[1] `https://shapenet.org`

[2] `https://m-selig.ae.illinois.edu/ads/coord_database.html`

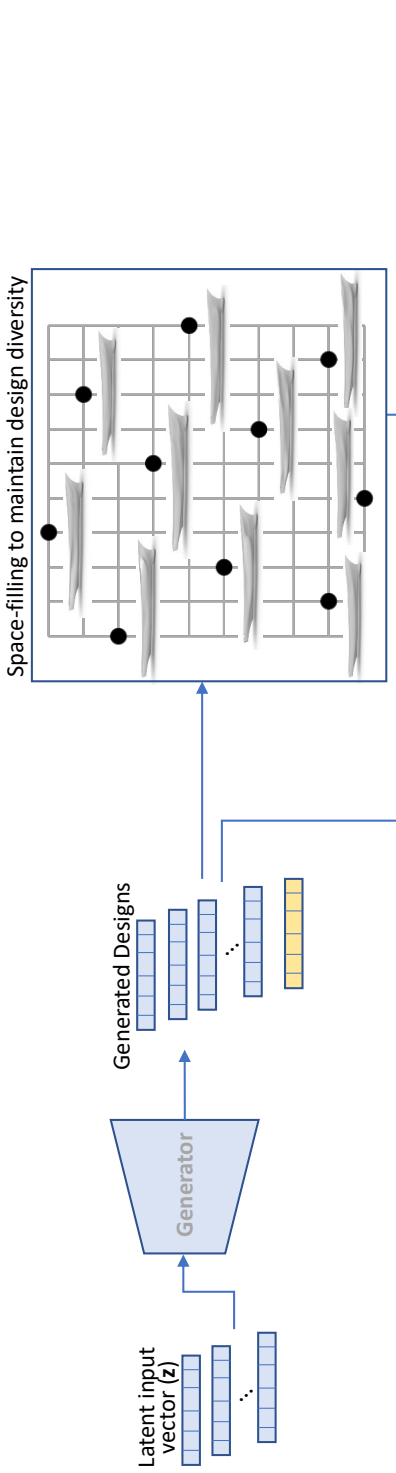


Figure 8.3: Overall architecture of ShipGAN.

This is probably why so far in ship design research, SciML models are implemented on a specific design type whose variations are created synthetically using a baseline parameterisation. However, in such cases, new hulls are generally slight variations of the parent hull (baseline design). Therefore, if GANs were trained on a specific ship-hull type using a similar baseline variation process, one could not expect significant novelties in generated designs. To overcome this hurdle and construct a sufficiently diverse and large dataset of existing ship-hull geometries, we extensively studied the pertinent literature on systematic hull form series, optimisation, and machine learning to extract all relevant hull types. This exercise resulted in consideration of systematic series, e.g. FORMDATA, and a variety of parent hull families from different ship types, e.g., KCS[3], KVLCC2[4], VLCC, JBC[5], DTC, and DTMB[6]), shown in Fig. 8.4, which are widely used in industry in academia.

Among the hulls in Fig. 8.4, the FORMDATA series is based on the statistical analysis of existing ships of various types and has been widely used for designing typical ships. It covers conventional wall-sided hull forms. The hull variation resulting from FORMDATA can provide approximately 5000 hull forms but of only three basic types of ship lines, referred to as U, N and V, which are generated by combining two series, A and F, for the after and fore parts, respectively. The shapes of these lines are varied systematically with respect to the midship section-area coefficient c_M and the block coefficients c_{BA} and c_{BF} of the aft and fore parts of the ship, respectively. Therefore, the dataset still needs to be more diverse. More importantly, if we add the previously mentioned 17 hull geometries within this dataset and use them for training our model, then there is a good possibility that their impact will be either zero or minimal. To overcome this issue, we created synthetic variations of the remaining hull in Fig. 8.4 based on the parametric approach discussed in [18]. These designs' length, beam and width are kept constant, and non-dimensional parameters, varying between 0 and 1, are used to create shape variations.

[3] http://www.simman2008.dk/KCS/kcs_geometry.htm
[4] http://www.simman2008.dk/kvlcc/kvlcc2/kvlcc2_geometry.html
[5] https://www.t2015.nmri.go.jp/jbc.html
[6] http://www.simman2008.dk/5415/combatant.html

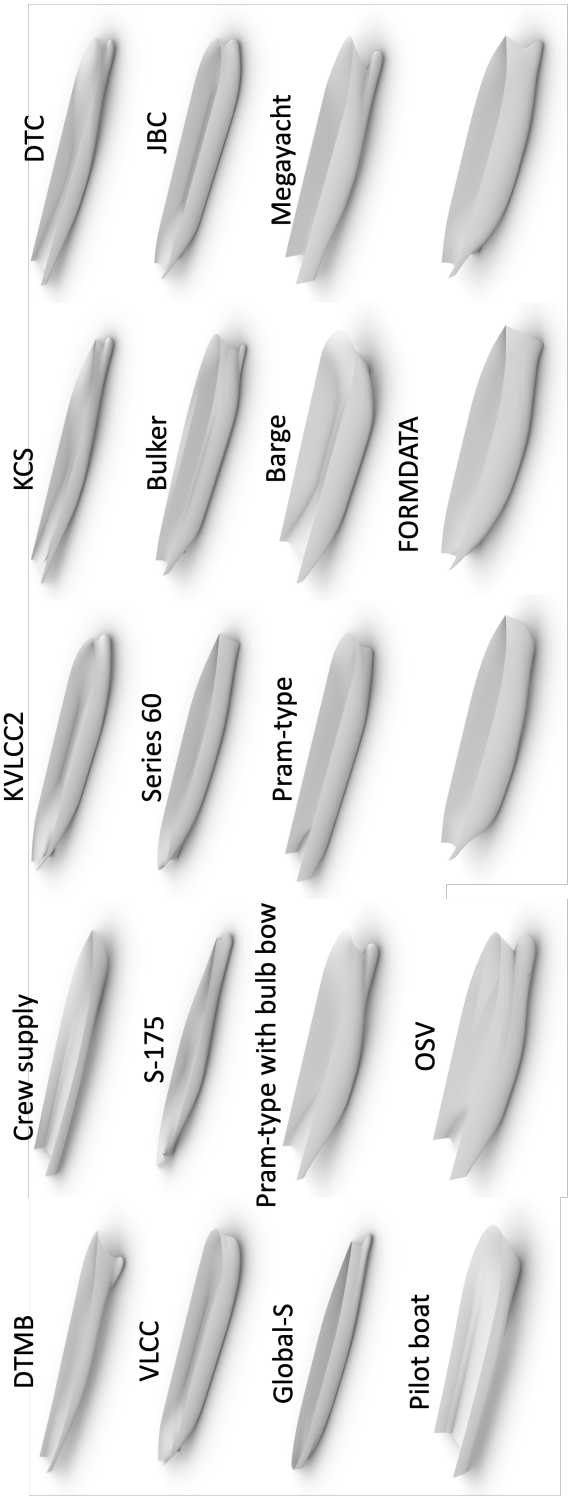


Figure 8.4: Parent ship hulls used for the training of ShipGAN model.

For some hulls, variation resulting from this parameterisation is shown in Figures 8.5 - 8.10. It can be seen that all the hull variations have plausible geometries with significant variations. We have 52,591 design variations used to train the ShipGAN model. The distribution of wave resistance (C_w) and volume of these designs are shown in Fig. 8.11.

8.3.2 Shape encoding for GANs

It is well known that deep learning models require datasets with vector inputs of fixed dimensions to extract meaningful features. This is relatively easy to achieve for natural language processing and/or vision/image processing, where these models originated. However, selecting suitable data encoding is a significant challenge when considering applications of deep learning models in 3D free-form shape processing. Free-form shapes, even when belonging to the same family, can have significantly different topology, structure, geometric parameterisation, and resolution; see Fig. 8.12 as an example of three ship hulls with significantly different geometrical representations and dimensionality. Therefore, we need to ensure that all shapes in the training dataset share the same underlying topology, representation, and resolution. All designs need to be converted into a common representation with a similar resolution at a prepossessing stage.

Signed distance function (SDF), voxels, point clouds and meshes are commonly used with satisfactory results for shape visualisation tasks in computer graphics and machine learning-based regression models for performance prediction [179]. However, in generative models, when the output is also a 3D shape, these approaches often result in the loss of local geometric features of the input shapes. More importantly, the resulting designs of such approaches commonly lack surface smoothness, which is crucial for several engineering analyses. In the case of ship hulls, both local features and surface smoothness are essential in appropriately evaluating the hydrodynamic performance of a ship hull. Although one can achieve a certain level of smoothness by increasing the employed resolution, this also increases the network complexity and memory requirements. A detailed discussion of such approaches with their advantages and disadvantages can be found in [179].

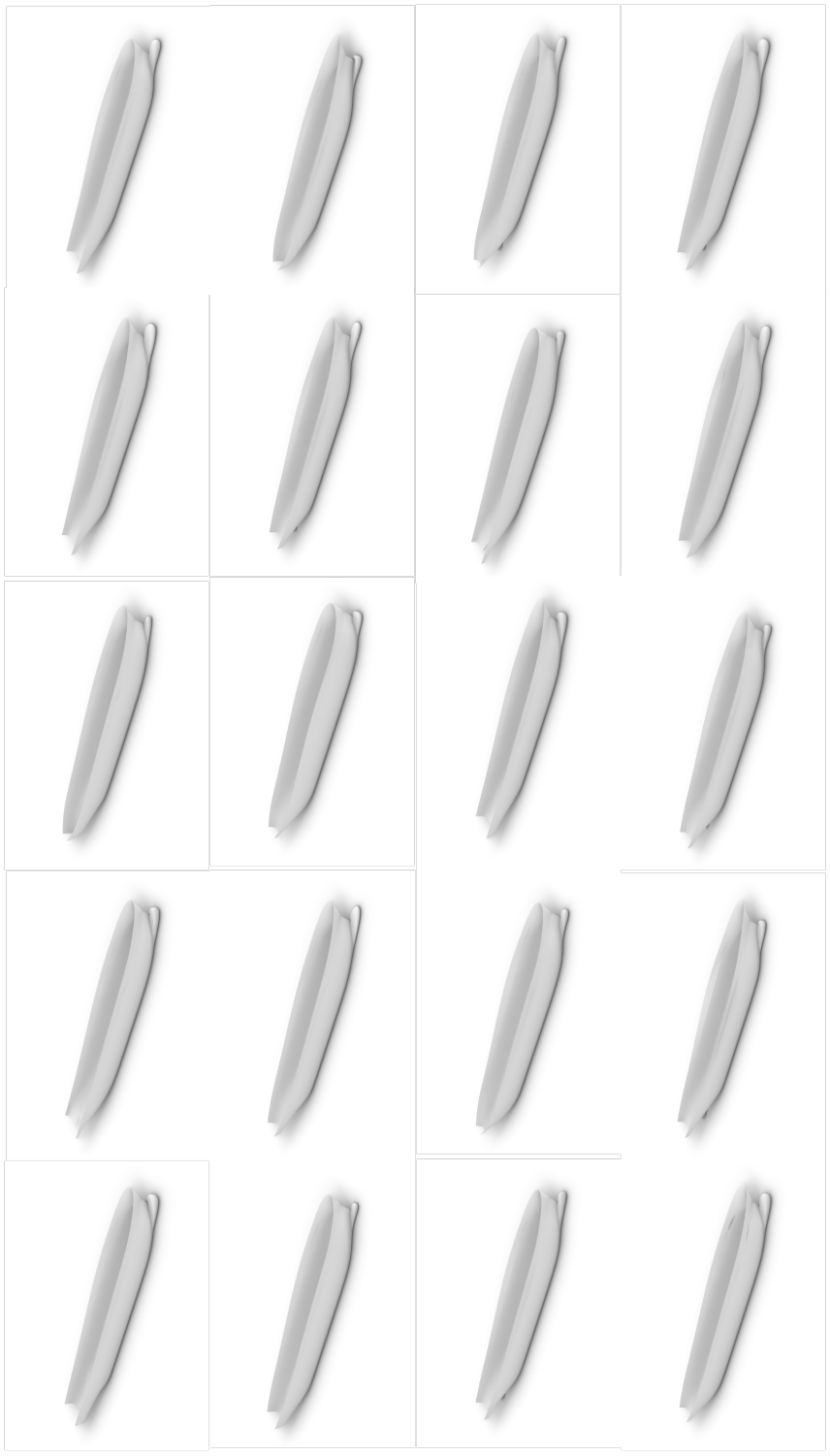


Figure 8.5: Some of the synthetic design variation of *Bulker* hull in Fig. 8.4 created for training ShipGAN.

Figure 8.6: Some of the synthetic design variation of *DTMB* hull in Fig. 8.4 created for training ShipGAN.

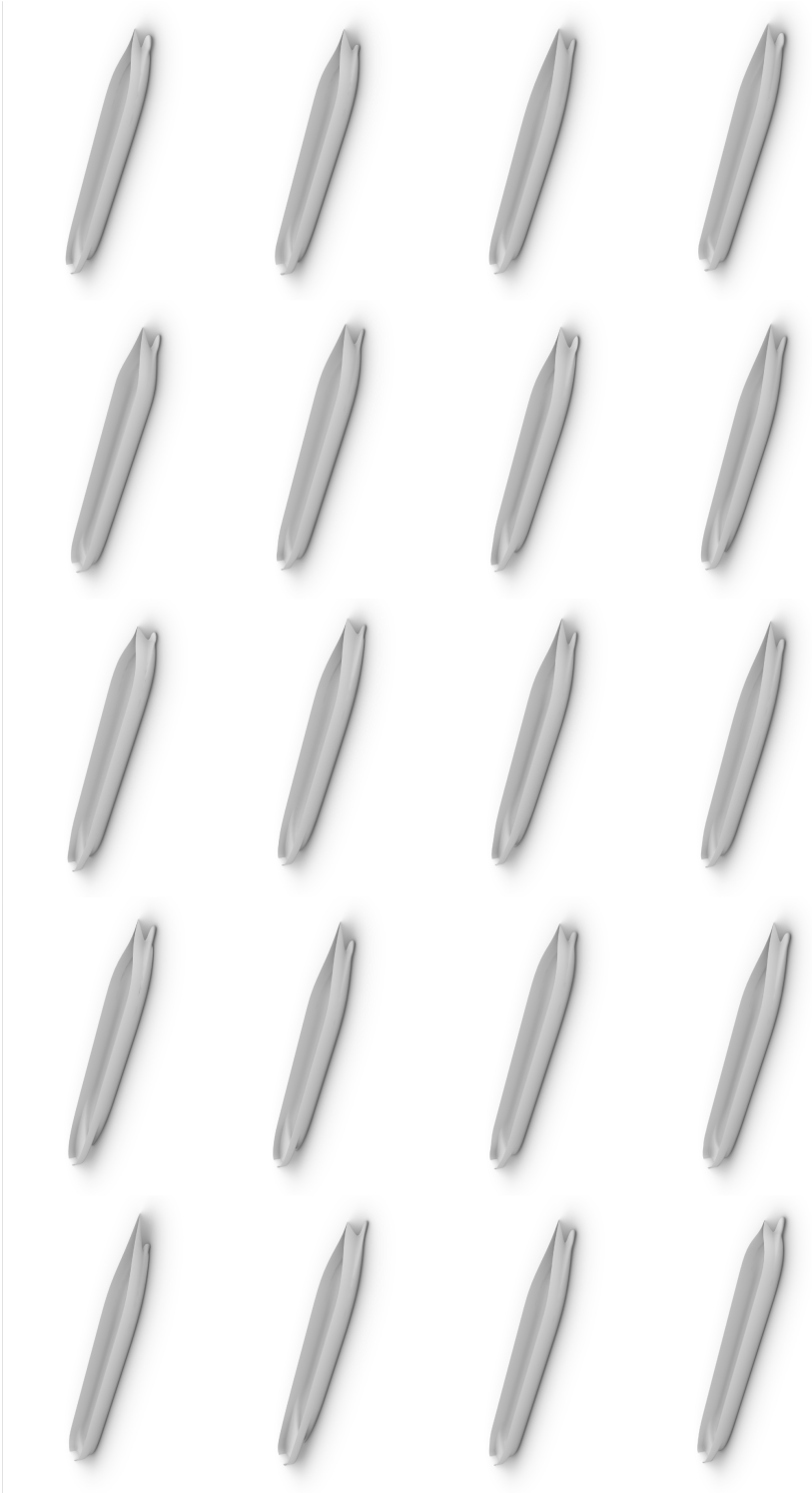


Figure 8.7: Some of the synthetic design variation of *Global-S* hull in Fig. 8.4 created for training ShipGAN.

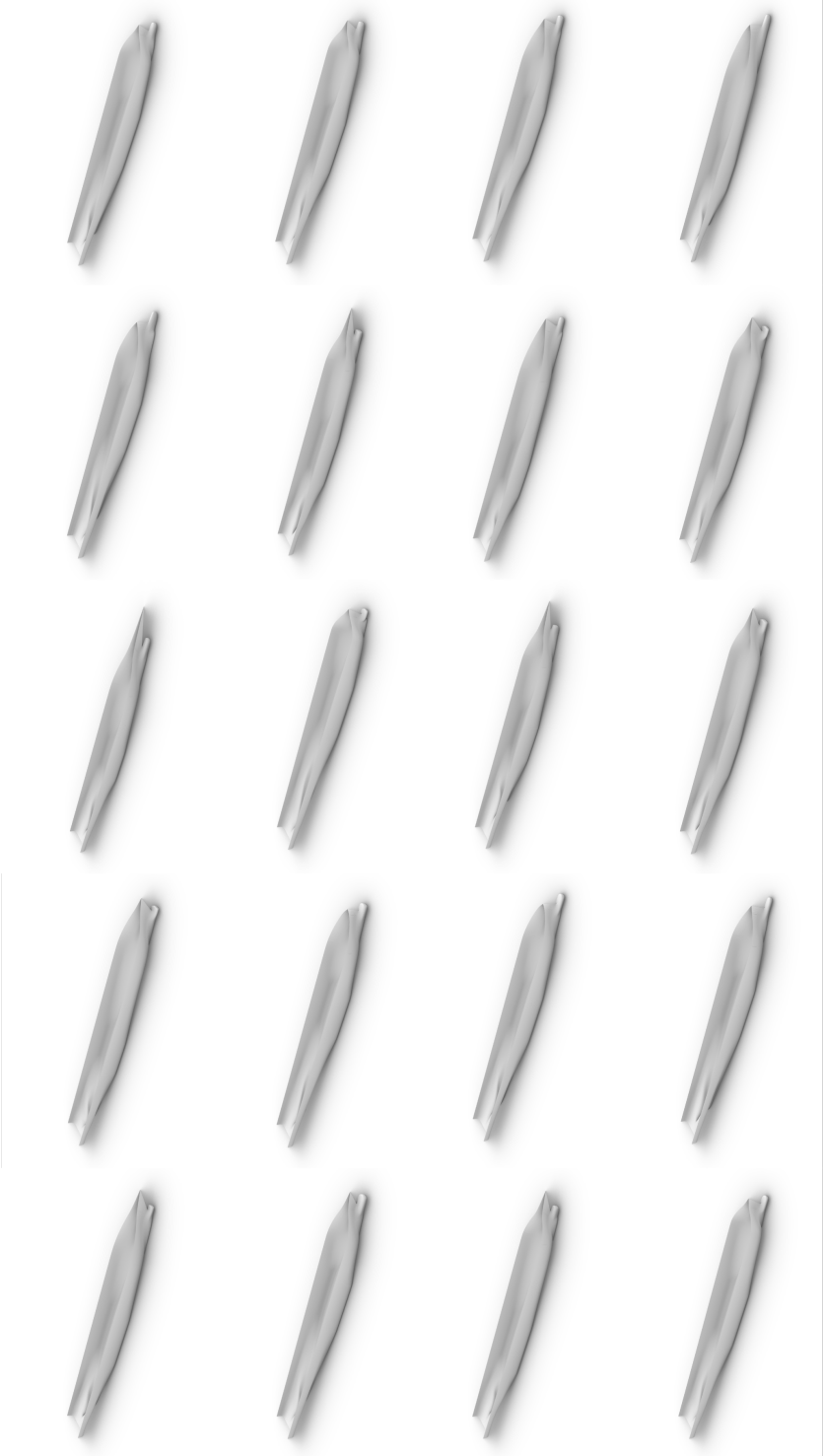


Figure 8.8: Some of the synthetic design variation of *KCS* hull in Fig. 8.4 created for training ShipGAN.

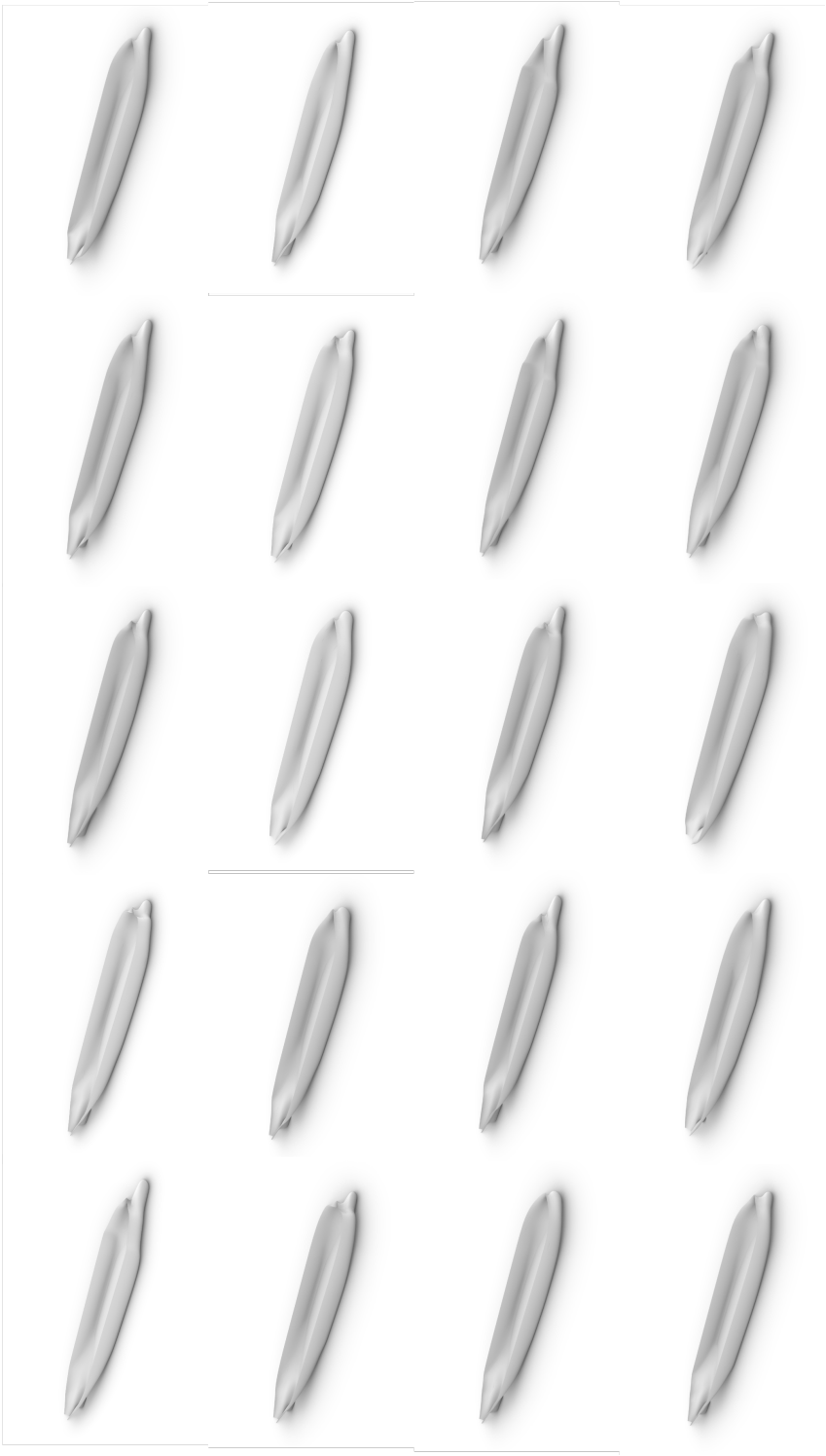


Figure 8.9: Some of the synthetic design variation of *KVLCC2* hull in Fig. 8.4 created for training ShipGAN.

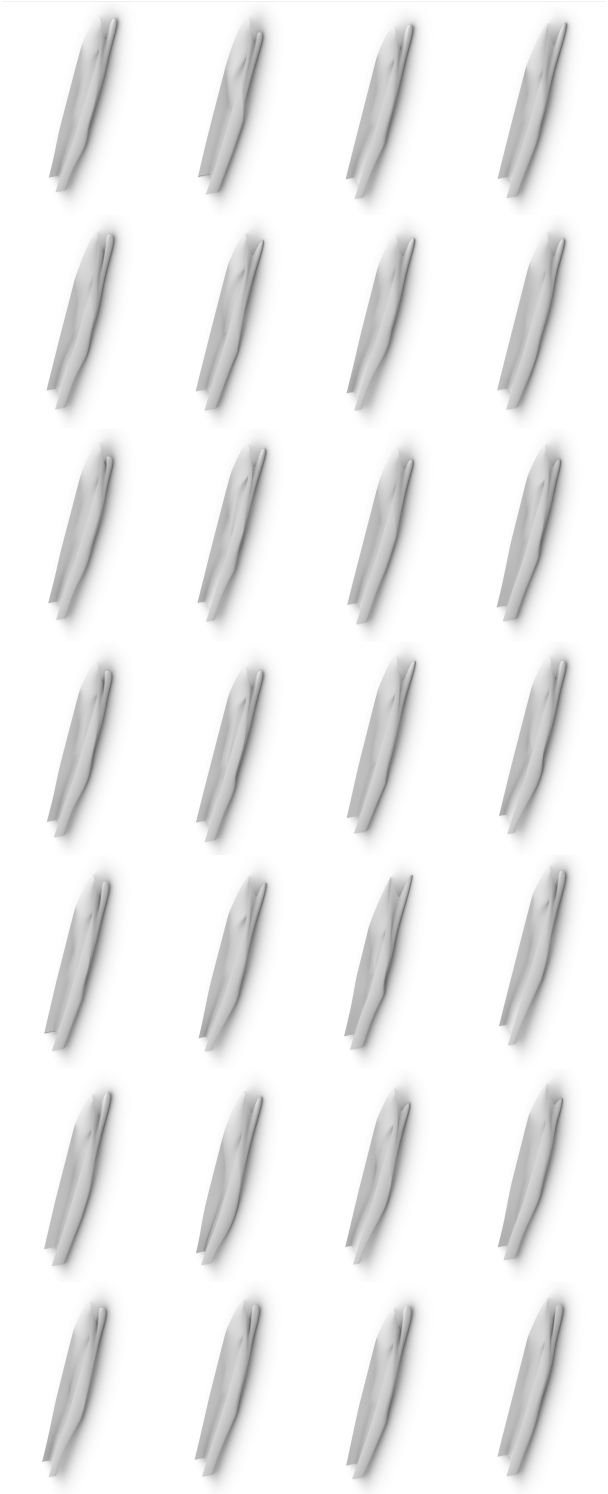


Figure 8.10: Some of the synthetic design variation of *Megayacht* hull in Fig. 8.4 created for training ShipGAN.

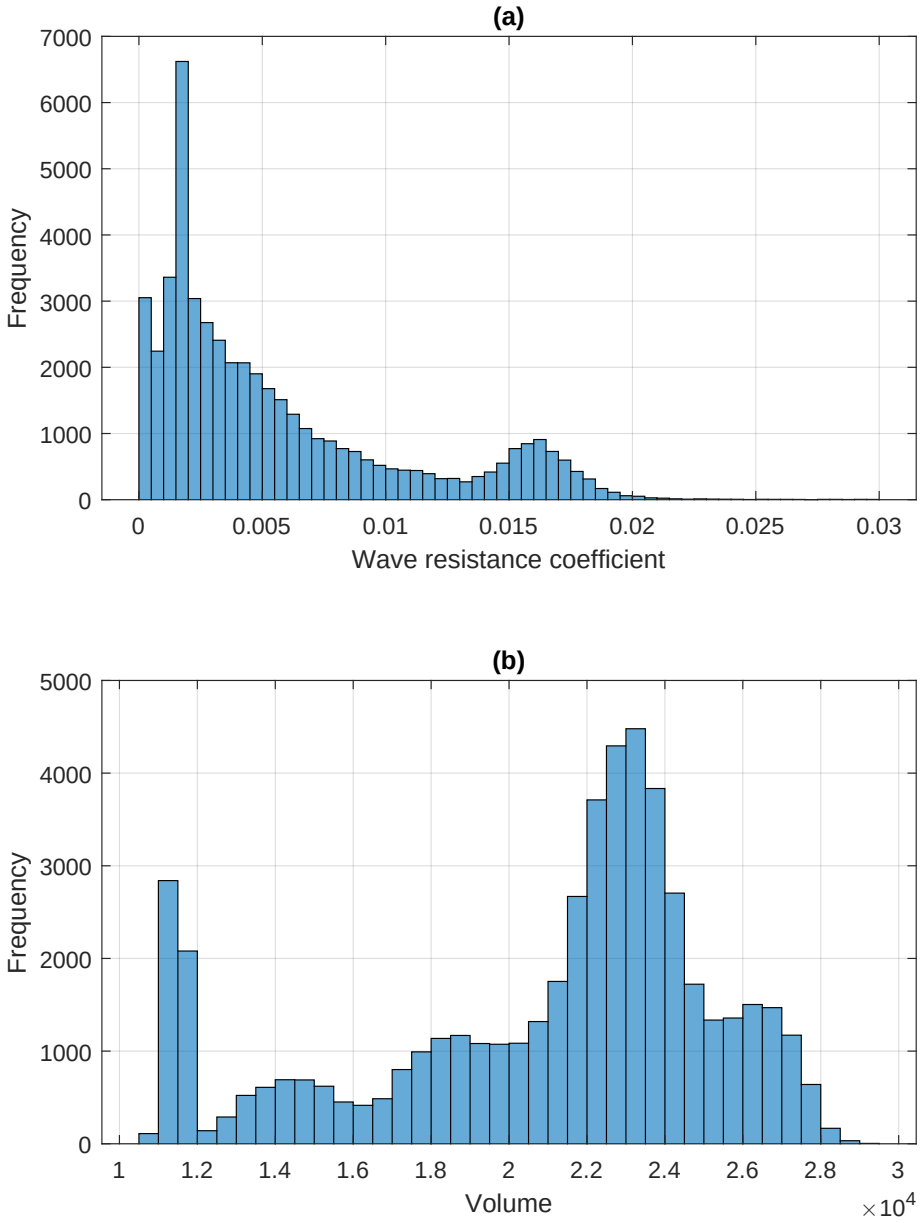


Figure 8.11: distribution of wave resistance coefficient and design volume in the training dataset of ShipGAN.

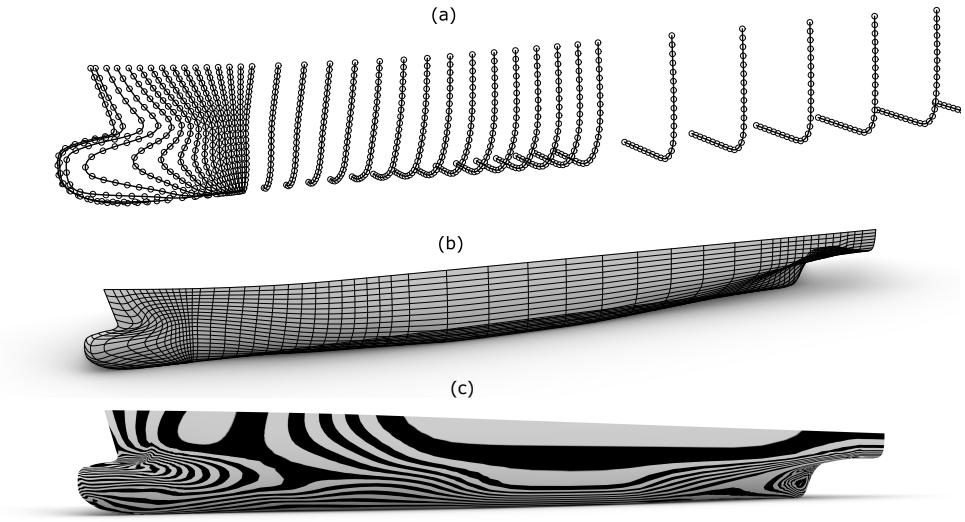


Figure 8.12: Example of three ship hulls with different surface structure and parameterisations: the DTMB hull is constructed with a single NURBS surface, whereas the KCS and S-175 are composed of several NURBS surface patches with a significantly different number of control points.

NURBS-based surface representations are quite common among ship hull designers as they provide the most accurate and versatile mathematical description of design geometry and are thus favoured in the pertinent industry. As mentioned before, DCGAN models require fixed dimensional vectors as input, and therefore a common fixed description is needed. However, especially when the dataset comprises different design classes, converting all of them into a common NURBS representation is not a trivial task, especially for 3D shapes.

In summary, any approach used for the construction of a 3D dataset for SciML training should:

1. represent all shapes with the same resolution;

2. capture both local and global geometric features of the shape;

3. maintain geometric similarity between the original and reconstructed shapes;

4. satisfy the above conditions with a relatively low resolution (e.g., with few mesh elements) to avoid redundancies and reduce the model's overall complexity.

In traditional and even modern ship design, the body plan (BP) is probably the most useful representation of the ship's hull lines. If appropriately constructed, it can be used, along with the basic reference lines, to develop the remaining ship lines plans, i.e., the profile plan and half breadth plan of the ship. Therefore, a body-plan-inspired approach can encode the geometric information in a uniform and consistent manner in a ship hull. Our approach is based on the intuitive arrangement of mainly transverse planes along the length of the ship hull so that all critical features of the hull surface are captured. More importantly, once a new design is generated from the GAN model, we can reconstruct a smooth and fair hull surface with sufficient accuracy and relative ease. The basic steps of our implementation are summarised below and illustrated in Fig. 8.13 for the KCS hull case.

1. Assume the axis-aligned bounding box shown in Fig. 8.13(b) with L, B, and D its longitudinal, transverse, and vertical dimensions, respectively, and the ship hull placed as shown in the same figure.

2. Convert into a non-dimensional representation using the bounding box length, i.e., the resulting principal dimensions $\bar{L}, \bar{B}$, and $\bar{D}$ of the scaled hull and a bounding box will now become 1, $\frac{B}{L}$, and $\frac{D}{L}$, respectively.

3. Divide the hull into four parts using a non-uniform partition, $[0, 0.1, 0.3, 0.8, 1]$, which corresponds to the typical regions of different geometric variation for ship hulls in the longitudinal direction; see Fig. 8.13(c). The intervals $P_1 = [0, 0.1]$, $P_2 = [0.1, 0.3]$, $P_3 = [0.3, 0.8]$, and $P_4 = [0.8, 1]$ correspond to the bow, fore transition, wall-sided (midship), and stern parts, respectively.

4. Assuming that m is the overall number of ship lines (curves) used to describe each ship hull in the dataset, divide this number equally to the 4 identified regions, i.e.,

$\frac{E}{4}$ lines for each region, P_1, P_2 P_3 and P_4. This arrangement generates a dense line description in areas with abrupt geometrical changes (P_1, P_2, and P_4) and a rather sparse representation for the region with an almost constant cross-section; see Fig. 8.13(d).

The employed cross sections (CSs) used in our encoding for P_2, P_3, and P_4 correspond to intersections of the ship hull surface with transverse planes, i.e., planes perpendicular to the longitudinal direction. However, CSs in P_1 are generated by a family of planes gradually "rotated" from the transverse to the longitudinal orientation as shown in Figs. 8.13(h-j). This approach is implemented to avoid disconnected cross-sections resulting from intersections of the bulbous bow area with transverse planes. In more detail, the following steps describe the construction of CSs in P_1:

1. Create the deck curve D_{P_1} of the hull part in P_1 and divide it into $\frac{E}{4}$ equally-spaced points; see Fig. 8.13(e)).

2. Define as p_{int} the upper edge point of the intersection line of the longitudinal plane of symmetry and the transverse plane at $\bar{L} = 0.1$. Using the line segments defined by p_{int} and each of the identified points on D_{P_1}, generate $\frac{m}{4}$ planes intersecting the ship hull; see Fig. 8.13(f).

3. Create CSs in P_1 by computing the intersections of the previously constructed planes and the ship hull; see Fig. 8.13(g).

4. Generate N points for each of the generated CSs in all regions; see Figs. 8.13(i,j).

8.3.3 Preparing geometric data for training

As previously mentioned, there are $n = 52,591$ designs in our shape dataset. Before training, all designs in this dataset are deconstructed using the previously described body-plane-based approach. For this deconstruction, we use $E = 56$ CSs, and each CS is divided

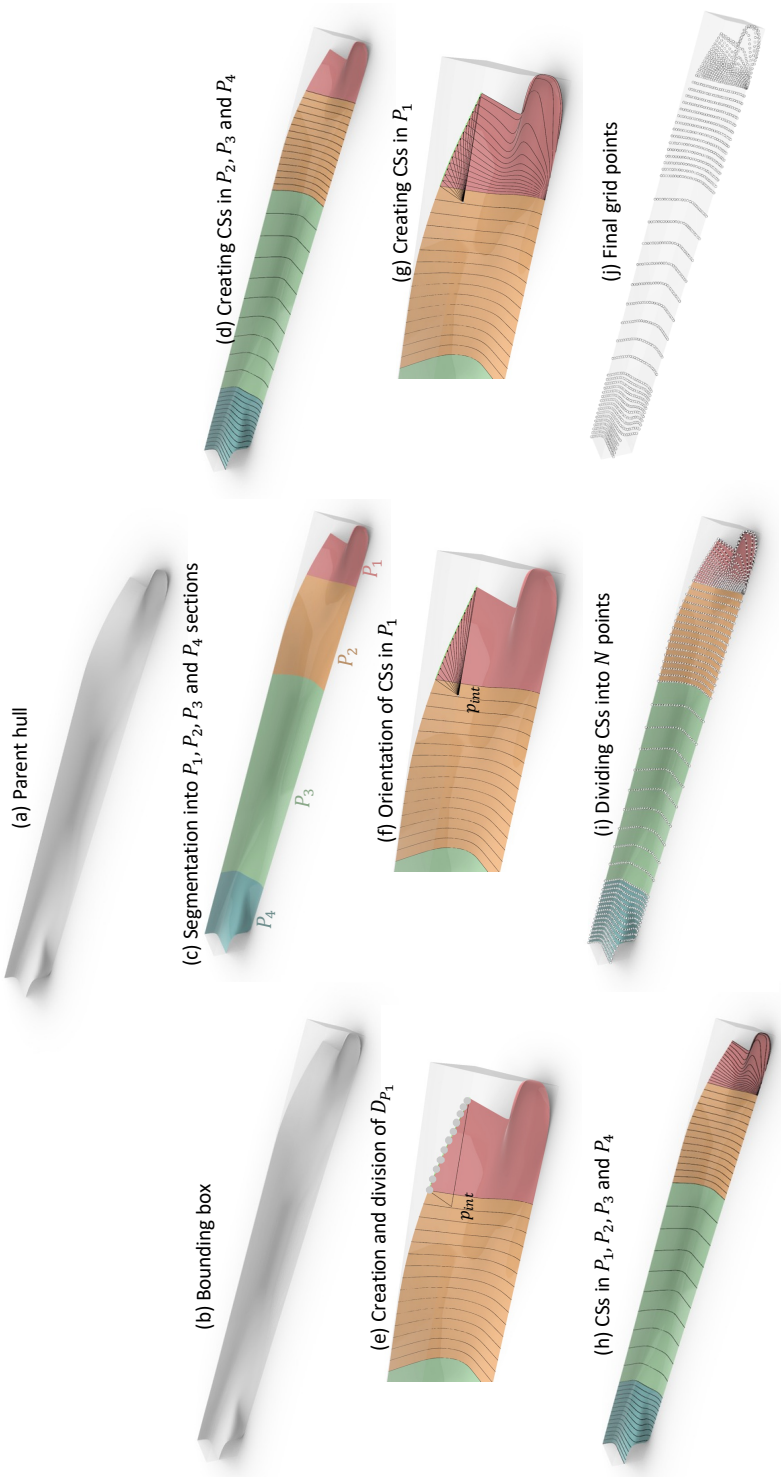


Figure 8.13: Steps of the proposed body-plan-based approach for extracting geometric information from ship-hull shapes.

into $N = 25$ points. Hence, the i^{th} design will be represented with $\mathbf{x}_i$, corresponding to 25×56 grid points. We have experimented with different grid resolutions, but, as indicated in Fig. 8.14, the employed, relatively low, resolution of 25×56 grid points provides sufficient surface reconstruction accuracy while preserving both local and global geometric features.

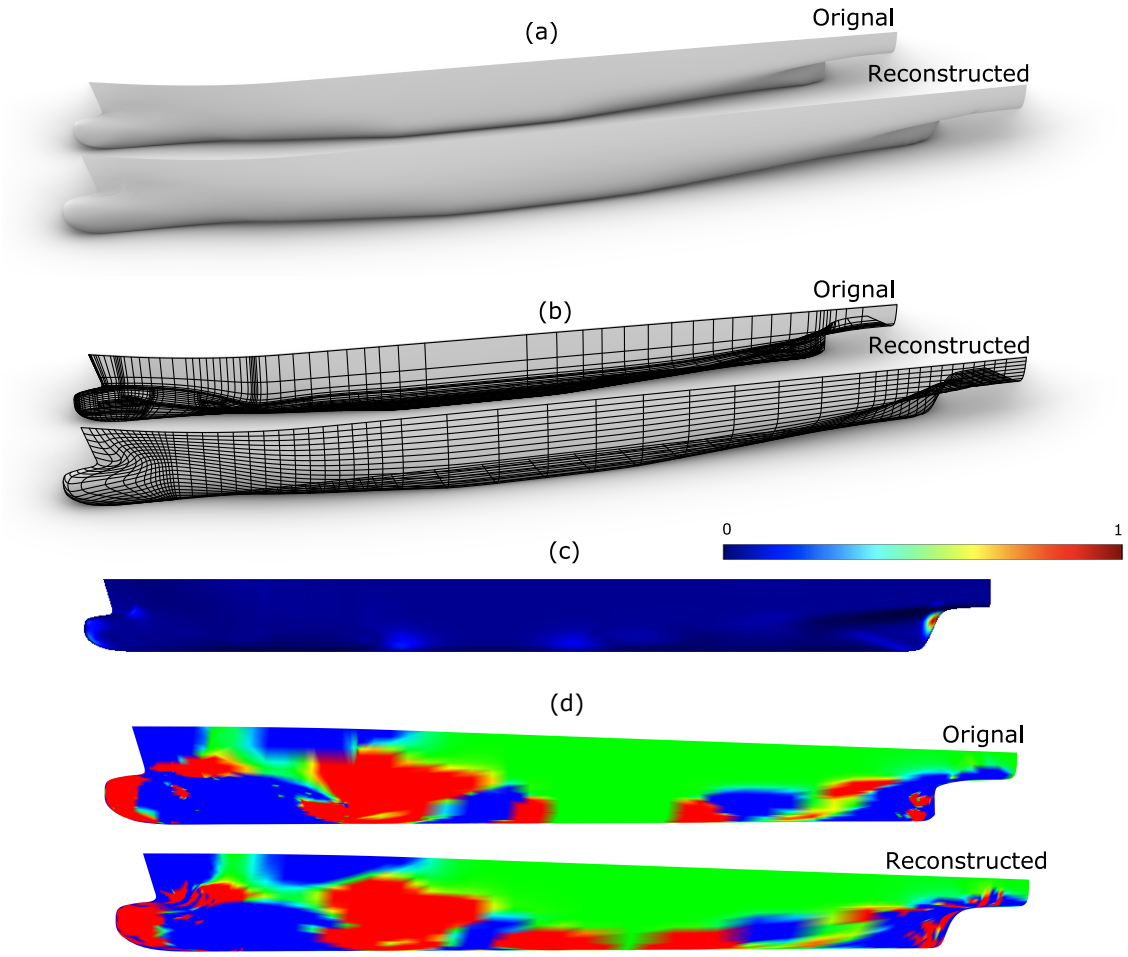


Figure 8.14: Comparison between the original KCS hull and its surface reconstruction from the grid points of the proposed body-plan-based approach. (a) Surface representations of the original and reconstructed hulls, (b) their geometric representation, comparisons in terms of (c) the one-sided Hausdorff distance [3], and (d) Gaussian curvature.

Finally, the x (longitudinal), y (transverse) and z (vertical) coordinates of the generated grid points are used to construct three $[25 \times 56]$ matrices as shown in Fig. 8.15. Hence, the

geometric representation/encoding of the shape dataset is materialised with $n = 52,591$ 3-tuples of $[25 \times 56]$ matrices.

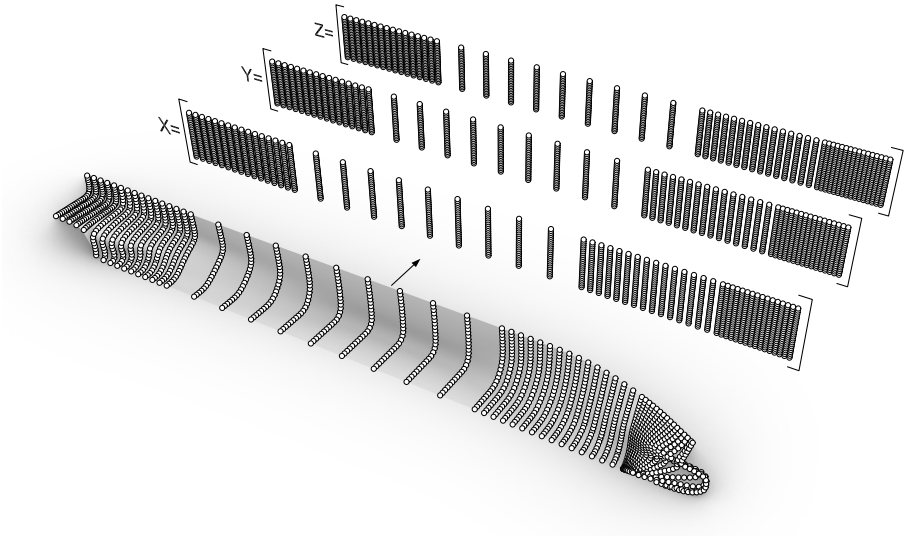


Figure 8.15: Illustration of transformation of grid points into training set's 3-tuples of input matrices.

8.3.4 Enhancing shape validity and diversity

As shown in Chapter 6 that using geometric moments along with the shape increases the chances of creating a large number of geometrically valid shapes, as adding moments gives a rich set of information about the geometry. More importantly, the existence of a strong correlation between physical QoI (i.e., wave resistance coefficient (C_w)) and geometric moments also induces the notion of physics in the extracted latent features (see §5.3.1 for further details). Thus, the resulting features have not only the ability to form a compact but also a physics-informed design, ensuring high-quality valid designs. We add GMs to the 4th order of each design into their grid point matrix. The last row of the matrix contains 32 components of the moments, and we add zeros in the remaining 25 elements to complete three $[25 \times 57]$ matrices for each design in the training dataset (see Fig. 8.16). Such matrices in x, y and z directions result in SST.

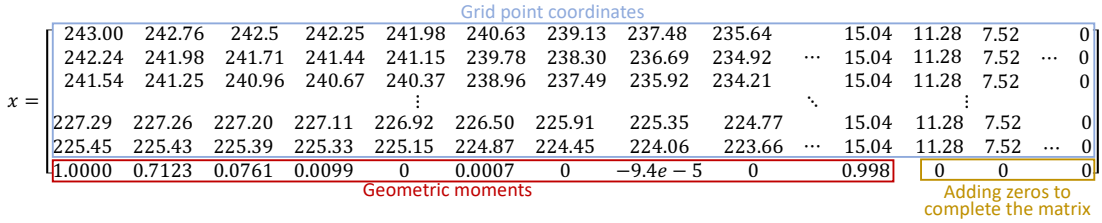


Figure 8.16: Structure of a matrix containing x coordinates of the grid points of a design in the training dataset.

If GANs are not adequately trained, then the resulting generator fails to map to the entire space of the training data. Thus, it can create only small subsets of designs, extensively lacking diversity [167]. This behaviour is prominent when the training dataset is composed of designs with different classes, like in the present case. Generators with such drawbacks can easily be analysed by visually inspecting a large set of resulting designs, as the designs with low diversity will be either identical, or a small subset of identical designs will be repeated many times.

Therefore, we introduced a space-filling criterion to enable ShipGAN to map latent features on the entire training space and enhance diversity. This criterion is implemented using Audze and Eglais [99] approach. It follows a physical analogy that molecules, designs in our case, in space exert repulsive forces on each other that lead to potential energy in space. These molecules are in equilibrium in case of minimum potential energy, which guarantees their uniform distribution over the entire design space. This criterion for m designs resulting from the generator is evaluated as

$$\mathcal{S} = \sum_{i=1}^{m-1} \sum_{j=i+1}^{m} \frac{1}{||\mathbf{x}_j - \mathbf{x}_i||_2^2}. \tag{8.3}$$

Here, $\mathbf{x}_i$ and $\mathbf{x}_j$ the pair of designs from the generator. Minimisation of $\mathcal{S}$ favours their uniform distribution of designs over the entire design space. For more details on space-filling, readers should refer to [13].

8.3.5 Loss function

The space-filling term (given in Eq. (8.3)) is added to the original loss function of the GAN given in Eq. (8.1), resulting in a new loss function written as:

$$\min_{G} \max_{D} \mathcal{L}_{adv}(D, G) + \Gamma_G \, \mathcal{S}. \tag{8.4}$$

Γ_G controls the weight of the space-filling term. Typically, as the training of GAN commences, it is more likely to generate unrealistic designs; therefore, at the start of the training, we set Γ_G equal to 0 and increase it during training so that ShipGAN focuses on learning to generate realistic designs at the early stage and takes space-filling into account later when the generator can produce more realistic designs. During the training Γ_G is set an escalating schedule proposed by [96], which is described as

$$\Gamma_G = \Gamma'_G \left(\frac{t}{T} \right)^p, \tag{8.5}$$

where Γ'_G is the value of Γ_G at the end of training, t is the current training step, T is the total number of training steps, and p is a factor controlling the steepness of the escalation.

Architecture of generator and discriminator

As mentioned at the beginning of the section, the generator, G, and discriminator, D, are materialised via deep convolutional neural networks whose structure is shown in Fig. 8.17. The discriminative network, D, consists of 6 convolutional layers and one input layer, which takes three $[25 \times 57]$ matrices of grid points (x, y and z coordinates) augmented with 4th order GMIs. A dropout layer, with a dropout probability of 0.5, succeeds the input layer to prevent over-fitting on the training data. This layer acts as a mask that randomly nullifies the contribution of some neurons toward the next layer. An activation layer follows each convolutional layer with a leaky rectified linear activation function (ReLU). The last convolutional layer uses a sigmoid activation function that calculates the probability of

a design being fake or real. For the second, fourth and fifth convolutional layers, batch normalisation is applied before the ReLU layer. The discriminator typically reduces data dimensions when assessing whether a design is real or fake in an operation that resembles downsampling when dealing with images. This downsampling in D is performed with strides of different padding sizes instead of the common pooling layer, as strides tend to improve the accuracy and stability of the model; see [166].

The generator, G, is the transpose of D and comprises 5 transposed convolutional layers, along with an input, projection and reshape layer. The input layer takes a randomly sampled $\mathbf{z}$ from a given distribution and feeds it to the "project and reshape" layer. Apart from the last layer, each convolutional layer is followed by batch normalisation and ReLU. The last convolutional layer of G has an activation layer with a hyperbolic tangent function to ensure an output value between -1 and 1, generating the normalised $[25 \times 57]$ matrices corresponding to our SST.

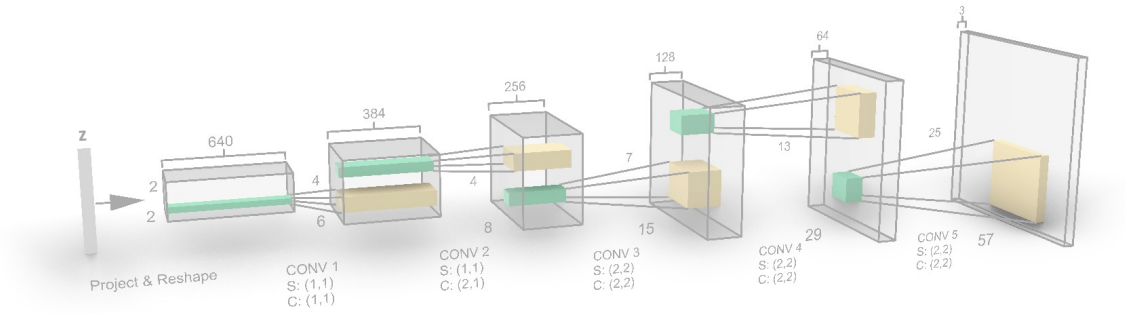


Figure 8.17: Convolutional architecture of the generator used in shipGAN.

This architecture resulted from systematic experimentation described in §8.4 and secures an adequately stable and smooth training procedure; additional details about the selection process and possible enhancements are given in §8.4. Model training is performed with the Adam gradient descent algorithm on a PC with dual 24-core 2.7GHz Intel® Xeon® 6 Gold 6226 CPU, NVIDIA Quadro RTX 6000 GPU and 128GB of memory, using

the following settings: number of epochs = 500; minimum batch size = 128, learning rate = 0.0002 and gradient decay factor = 0.5. Generator and discriminator networks employ 9.7 and 9.6 million learnable parameters, respectively.

Size of the input feature vector z

Unlike other techniques, such as principal component analysis (PCA) and others, the determination of the latent vector's ($\mathbf{z}$) size can be challenging in GANs. An inappropriate size for $\mathbf{z}$ can easily result in mode collapse issues, i.e., the generator learns to map several different $\mathbf{z}$ vectors to the same output [167]. Especially when $\mathbf{z}$ is small, the possibility of the generator's failure to cover the entire training dataset distribution increases, and it may produce many invalid designs and/or designs with minimal diversity. Obviously, a larger $\mathbf{z}$ may resolve this, but not without cost, since large vectors correspond to high-dimensional design spaces when performing shape optimisation, which increases the computational complexity of the entire simulation-driven design pipeline. Therefore, for estimating a sufficient but not redundant size of $\mathbf{z}$, we perform PCA and use the number of eigenvalues required for achieving a target variance as a reasonable estimation of the initial size of $\mathbf{z}$.

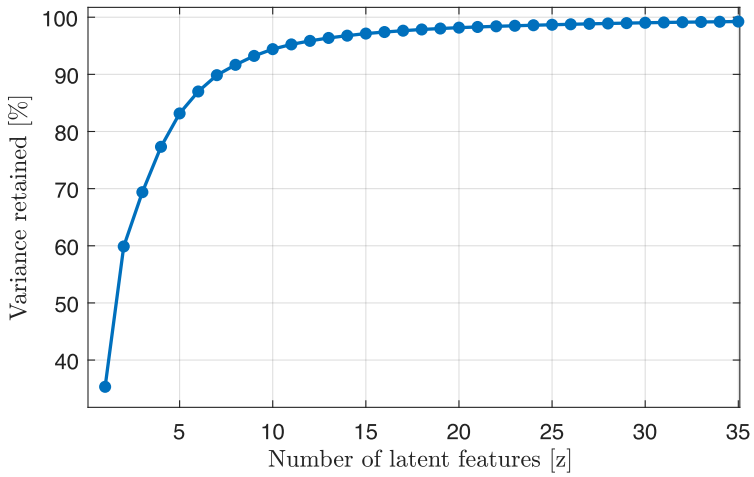


Figure 8.18: Percentage of variance retained versus size of $\mathbf{z}$.

As it can be easily seen from Fig. 8.18, 30 latent features in $\mathbf{z}$ can capture 99% of

geometric variance. We, therefore, set the initial size of $\mathbf{z}$ to 30 and then reduce it iteratively while measuring the diversity, novelty, and maximum mean discrepancy (MMD) [166] of generated designs. The variety and novelty are estimated with the sparseness at the centre (SC) [180] and the novelty score described in [96], respectively. The MMD metric is evaluated using Eq. (8.6) below, which measures the similarity between the distribution of designs in the training dataset and designs resulting from the generator. A high value of the MMD means that the generator cannot completely cover the design space in the training dataset, which may indicate a mode collapse issue. We may also note here that as GAN incorporates nonlinear layers, it should be able to capture the variability and nonlinearity in the training dataset with fewer latent variables compared to PCA. Thus, the initial size 30 can also be considered as an upper bound for the size of $\mathbf{z}$.

$$\text{MMD} = \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n} k\left(\mathbf{x}^i, \mathbf{x}^j\right) + \frac{1}{m^2}\sum_{i=1}^{m}\sum_{j=1}^{m} k\left(\mathbf{x}^i_{GAN}, \mathbf{x}^j_{GAN}\right) - \frac{2}{nm}\sum_{i=1}^{n}\sum_{j=1}^{m} k\left(\mathbf{x}^i, \mathbf{x}^j_{GAN}\right), \tag{8.6}$$

where $\mathbf{x}$ and $\mathbf{x}_{GAN}$ correspond to designs in the training dataset and designs generated from the generator, respectively, with n and m being the corresponding total numbers of the two sets of designs. Finally, k is a radial kernel function defined as

$$k(\mathbf{x}, \mathbf{y}) = \exp\left[-\frac{||\mathbf{x} - \mathbf{y}||_2}{2\theta^2}\right], \tag{8.7}$$

with $\theta = 0.1$.

We evaluate SC and novelty metrics using Eqs. (8.8) and (8.9), respectively. The SC measures the average distance of the centroidal design, $\mathbf{x}^{centroid}_{GAN}$, to the m designs resulting from ShipGAN. In contrast, novelty evaluates how different newly generated designs are from the designs in the training dataset, $\mathcal{X}$. It is estimated first by finding the nearest distance between the ith new design, $\mathbf{x}^i_{GAN}$, and all n designs in $\mathcal{X}$, and then by averaging all of those m nearest distances.

$$SC = \frac{1}{m}\sum_{i=1}^{m} ||\mathbf{x}_{GAN}^{centroid} - \mathbf{x}_{GAN}^{i}||_2 \tag{8.8}$$

$$Novelty = \frac{1}{m}\sum_{i=1}^{m} \min_{\mathbf{x}^j \in \mathcal{X}} ||\mathbf{x}_{GAN}^{i} - \mathbf{x}^{j}||_2. \tag{8.9}$$

Here, $\mathbf{x}^j$ are the designs in the training dataset, $\mathcal{X}$.

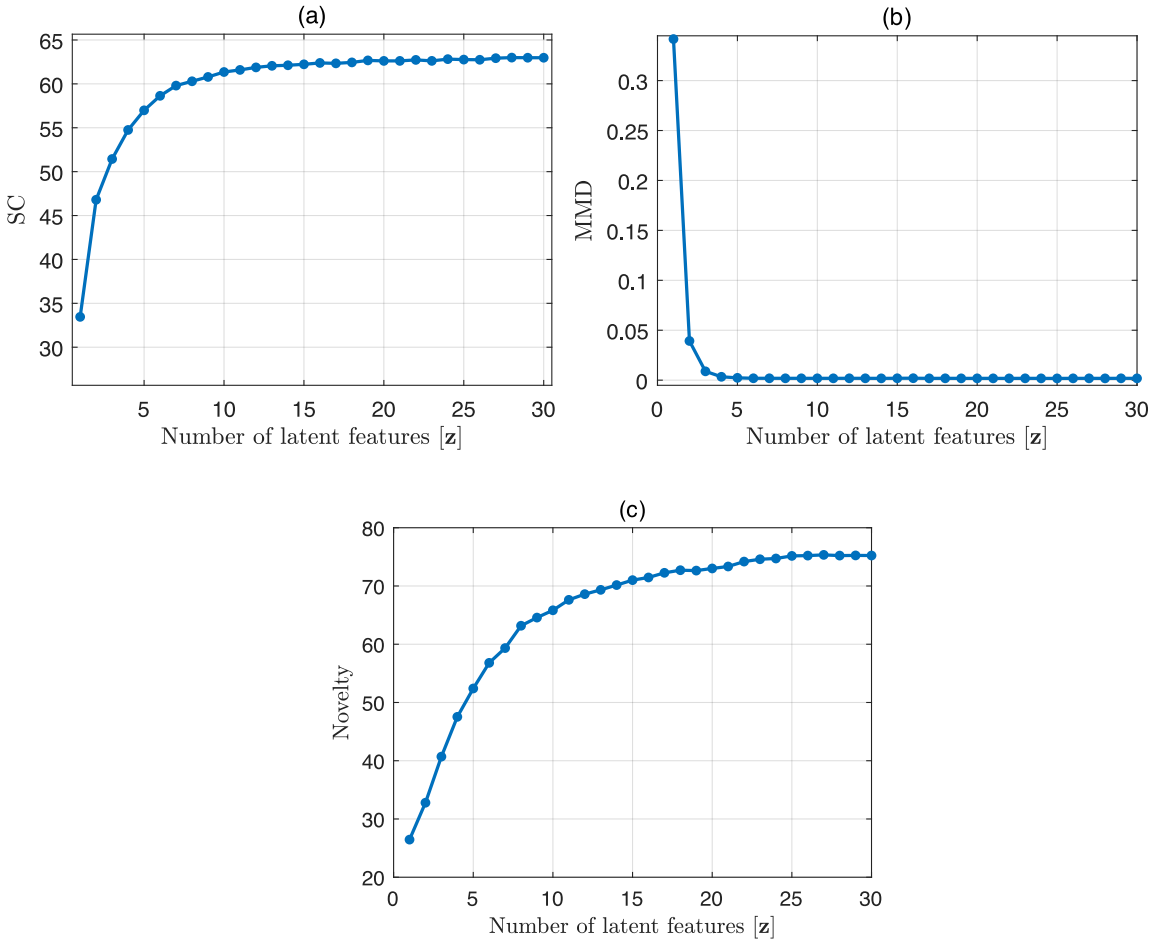


Figure 8.19: Plots depicting the value of (a) SC, (b) MMD and (c) novelty metrics evaluated using Eqs. (8.8), (8.6) and (8.9), respectively, versus the number of employed latent features.

We analyse the influence of latent space dimensionality against these three metrics in

Fig. 8.19. Higher values of SC and novelty generate diverse designs, while low values of MMD correspond to good coverage of the design space $\mathcal{X}$ by the generator. Figure 8.19 clearly indicates that as the number of latent features increases, diversity and novelty increase approximately up to the number of 20 features and then tends to plateau. In contrast, the MMD reduces rapidly and reaches a sufficiently low value with 5 features. These results indicate that 20 features is a well-balanced selection for the size of feature vector $\mathbf{z}$, and as it will be demonstrated subsequently, a generator trained with 20 features produces valid and physically-plausible designs.

8.4 Experiments: Design synthesis and optimisation

This section presents the process and experimentation results used to validate the appropriateness and efficiency of the proposed model.

8.4.1 Design reconstruction

After the training process has been completed, we use the generator of the trained model as a parametric modeller with 20 parameters ranging between -1 and 1, generating design in a 20-dimensional subspace $\mathcal{Z}$. For an input vector $\mathbf{z}$ sampled from $\mathcal{Z}$, the generator produces three $[25 \times 57]$ matrices corresponding to the x, y and z coordinates of grid points of a new design. Recall that the last row in these matrices corresponds GMIs; therefore, we remove this row from all three matrices to construct the final shape. The shape reconstruction using a NURBS surface of the new design is generated by first fitting a NURBS curve to the points of each cross section (CS); see Fig. 8.20(a). Then, the 3D surface representation is created by interpolating the reconstructed CSs with a bicubic NURBS surface using a skinning scheme (a.k.a. loft operation) as shown in Fig. 8.20(b). The resulting surface is smooth and fair with sufficient continuity, as indicated by using an isophotes mapping analysis (zebra stripes) on the reconstructed hull surface shown in Fig. 8.20(b)). The smooth transition of the zebra stripes on the surface indicates a smooth

and fair hull surface of C^2 continuity.

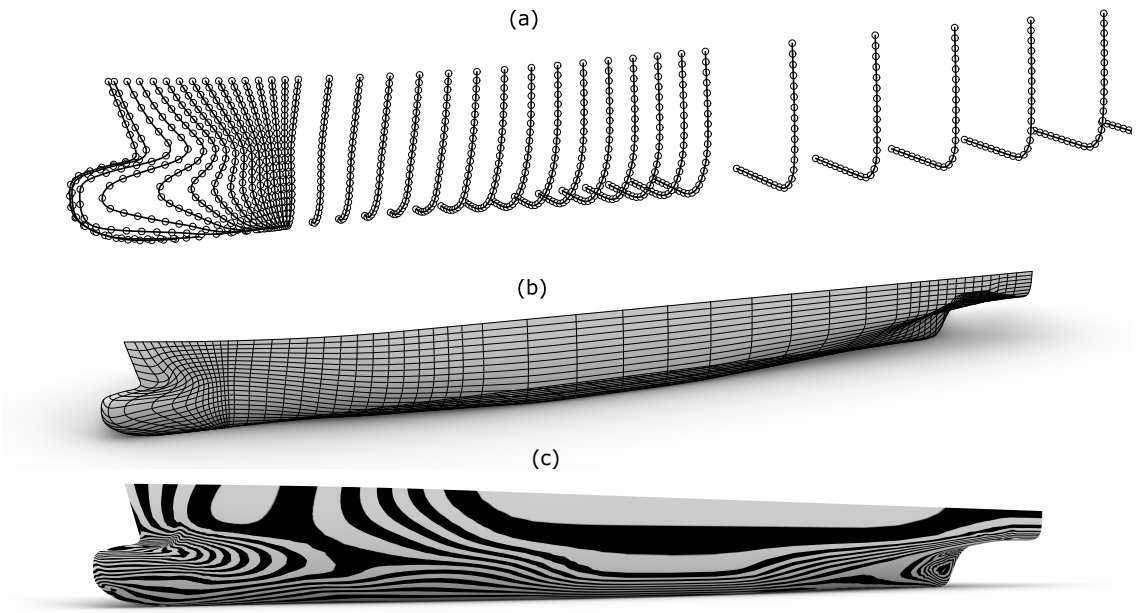


Figure 8.20: (a) Interpolation of points of CSs using cubic NURBS curves. (b) Construction of NURBS surfaces interpolating the curves with a loft operation. (c) Inspection of hull surface fairness using isophotes mapping analysis.

Indicative variations of the ship hulls generated using the ShipGAN model are shown in Fig. 8.21. From a visual inspection of these designs, a designer can easily conclude that these designs are physically valid and plausible with distinct geometric features and characteristics. One can also easily identify augmented features from the designs in the training dataset on several of the generated designs have . In Fig. 8.22, we depict three generated hulls from the ShipGAN model and the correspondence of their features to existing hulls. For example, the new design on the top right corner of Fig. 8.22 adopts features in the bow (green arrows), aft (grey arrows), and stern (orange arrows) regions, resembling to JBC, Megayacht and DTC parent hull features[7], respectively. This supports our claim that the proposed generic parametric model can generate hulls with diverse features from completely different ship hull types, which is one of the features existing

[7]see also Fig. 8.4

parametric modelling approaches in hull design largely lack.

8.4.2 Design validity and diversity

The geometric validity of designs resulting from the model is partially tested by searching for designs with self-intersecting geometries. We randomly sampled 30,000 designs over ten runs and searched for self-intersecting geometries. Interestingly enough, no self-intersections were found in any of the 300K tested designs. This is a strong indication that the ShipGAN model is a robust and efficient model, and these properties are attributed to its convolutional architecture, reliable training and inclusion of GMIs in the SST.

However, even though no self-intersecting geometries were detected, some of the ShipGAN-generate designs may be implausible from a practical point of view. Examples of such designs are shown in Fig. 8.23. Nevertheless, the possibility of receiving such designs is rather low as a visual inspection of large numbers of randomly sampled designs resulted in less than 1 out 70 instances with questionable designs. Even such designs can be eliminated by setting appropriate design constraints and/or employing the physical solver to rule out such designs during design optimisation.

We also use t-distributed stochastic neighbour embedding (t-SNE) [181] to further analyse the diversity and potential model collapses, i.e., closely clustered and/or identical designs from the generator or insufficient coverage of the design space. t-SNE is a statistical method for visualising high-dimensional data by giving each data point a location in a 2D or 3D map and can provide some indication of the distribution of designs. From Fig. 8.24, it can be seen that newly generated designs cover well the entire convex hull enclosing the designs in the training dataset. It should be noted that the topology of the t-SNE plot, more precisely the distance between the cluster, their size and orientations, may not have any physical meaning and therefore, in the present case, it's mainly used to visualise the distribution of generated designs within the training space. Moreover, as can be seen in the same figure, some of the new designs reside out of the convex hull, which according to [96], further indicates the ability of the generator to create novel designs. In summary,

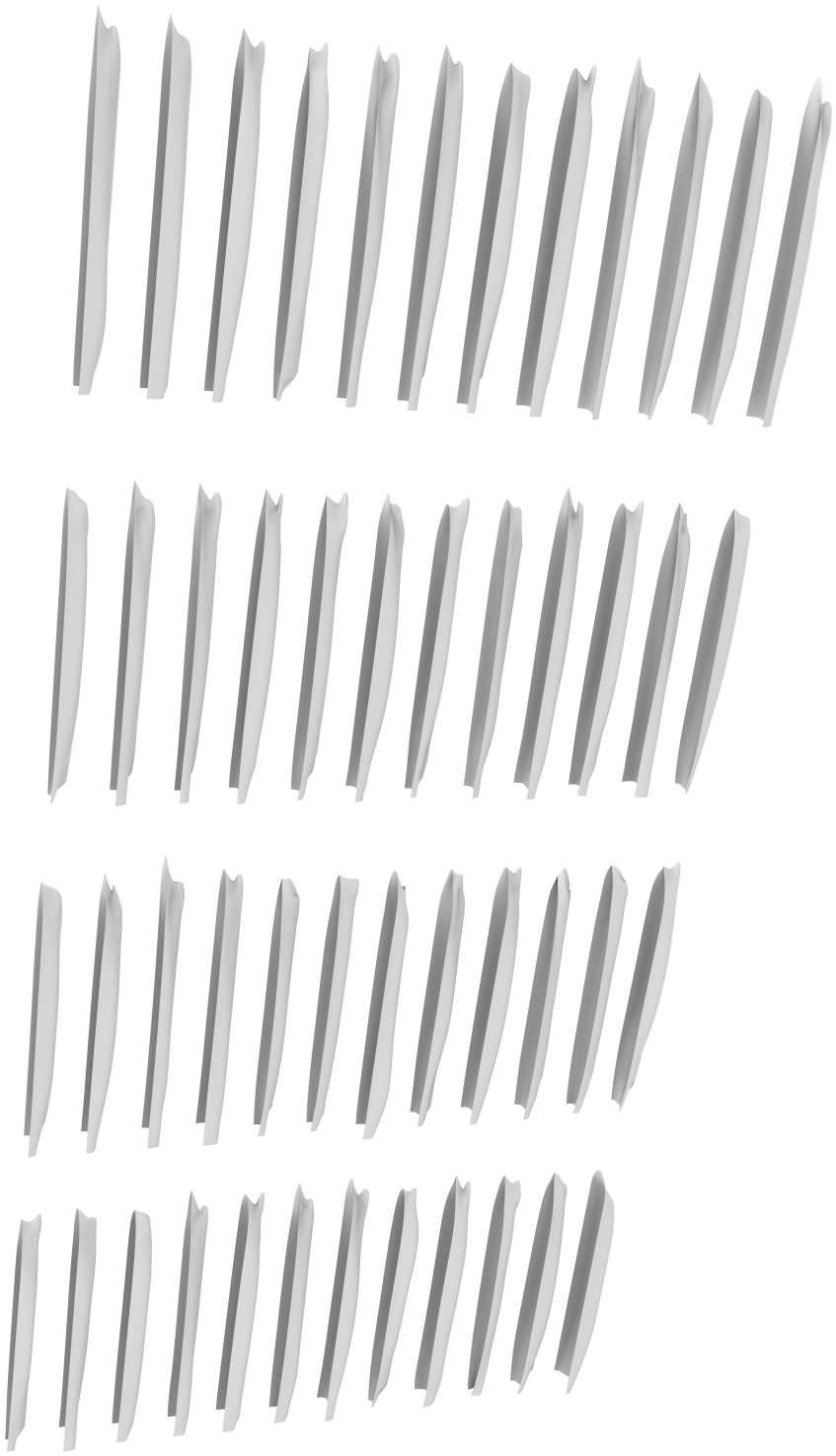


Figure 8.21: Design variations created with ShipGAN. Randomly sampled designs from $\mathcal{Z}$ and design variations resulting from changing each of the variables in $\mathbf{z}$ can be visualised at `https://youtu.be/ZIfmAs5-qFw` and `https://youtu.be/avlqOFxZP-s`, respectively.

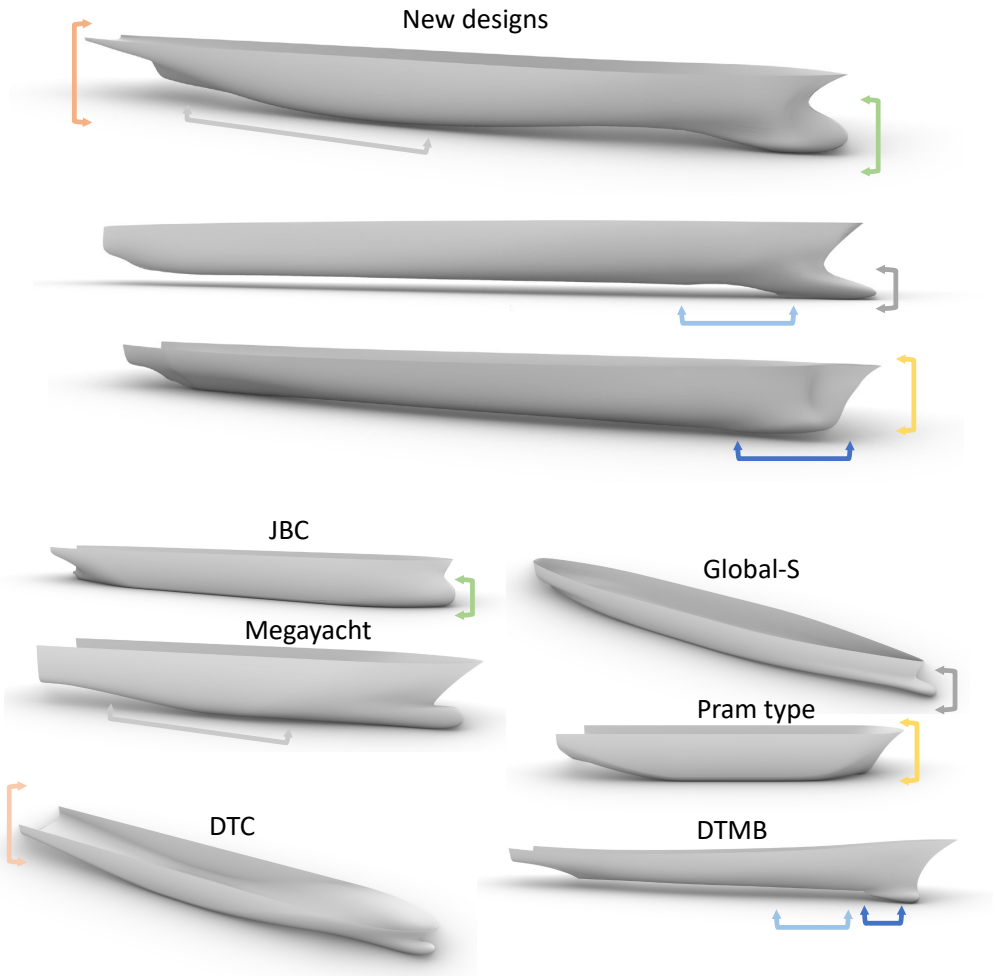


Figure 8.22: Examples of newly generated designs using ShipGAN adopting features from parent designs in Fig. 8.4.

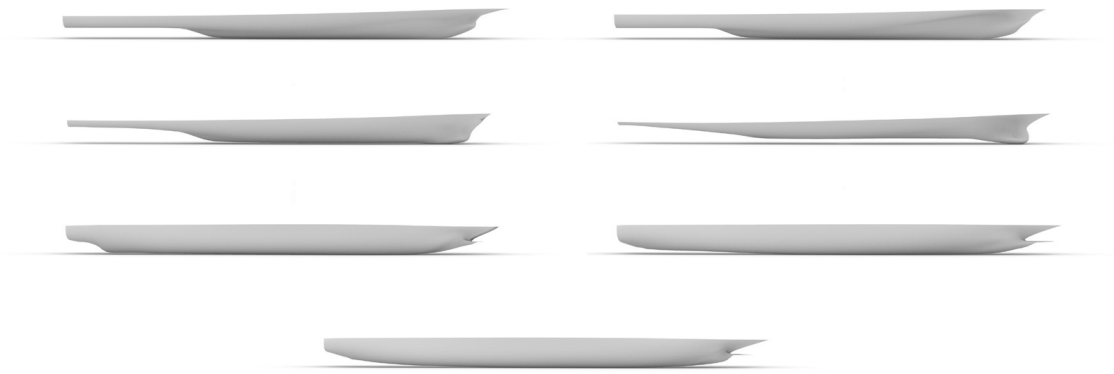


Figure 8.23: Example of implausible designs.

these results demonstrate that the parametric modeller resulting from ShipGAN is able to generate

1. designs similar to the training dataset (new designs overlap the existing ones),
2. designs with augmented features from different classes of design in the training dataset (new designs between the clusters), and
3. completely novel designs (new designs outside the convex hull).

Comparison with GAN

We finally compare ShipGAN with a GAN model trained with the exact same settings and architecture as ShipGAN but without space-filling and GMIs components to highlight their respective impact. We first evaluate the SC metric for both models, using 30,000 randomly sampled designs over ten runs (300K designs in total). The results of this experiment are shown in Fig. 8.25. It can be easily seen that the ShipGAN model shows significantly higher diversity and novelty compared to the GAN. We also conducted a t-test to see if there exists a significant difference between the diversity values. The $p-$values resulting from this test are $3.7354E-09$ and $2.1315E-09$, respectively, which are lower than 0.05, indicating a significant difference.

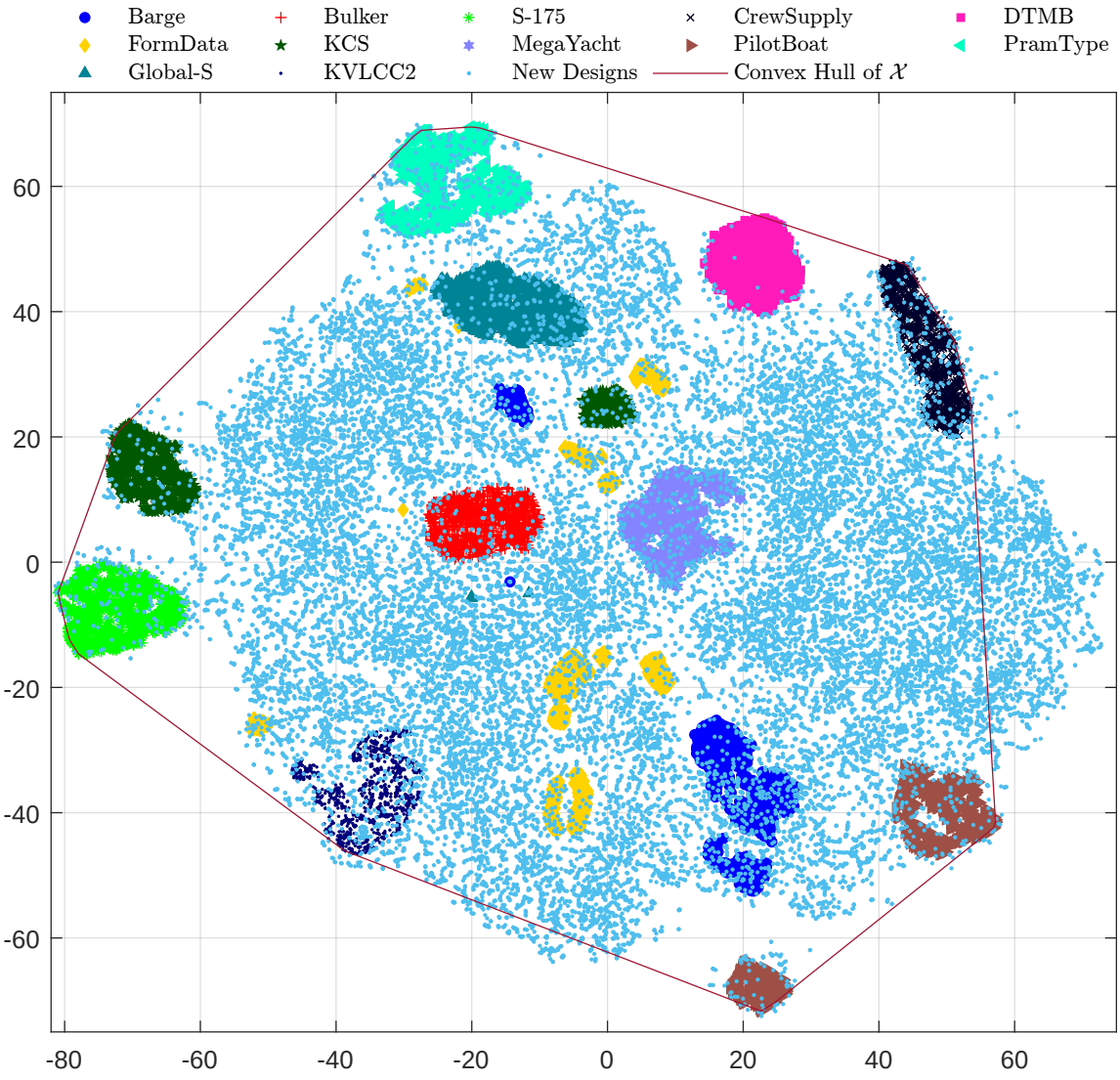


Figure 8.24: t-SEN plot of some design in the training data and newly generated designs from the ShipGAN model.

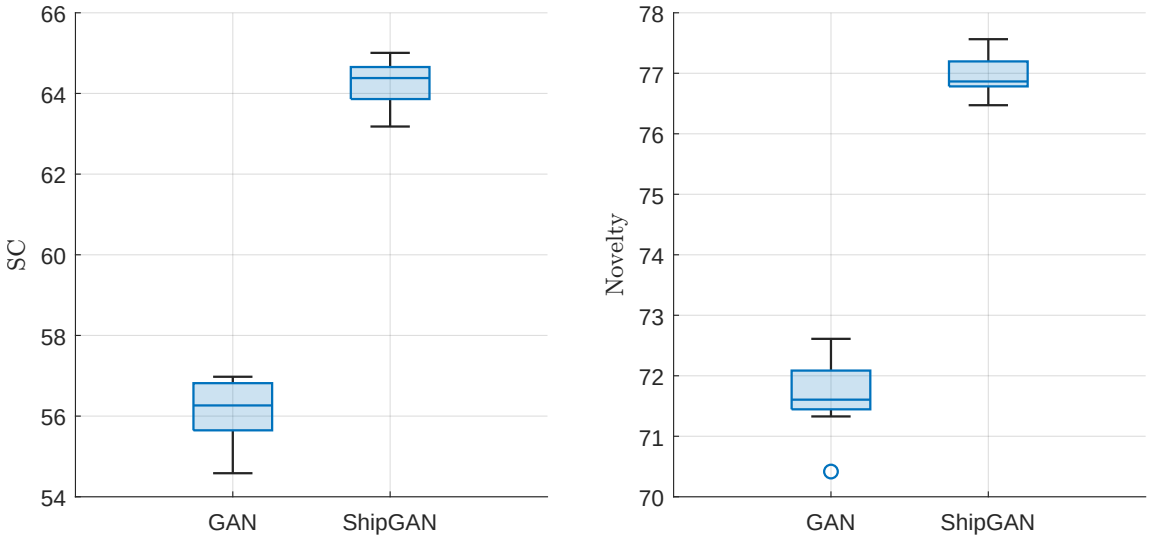


Figure 8.25: (a) Diversity and (b) novelty of designs created with the generator of GAN and ShipGAN.

Furthermore, we also analysed the ability of GAN to produce valid designs, i.e., designs with non-self-intersecting surfaces, by once again sampling 30,000 designs over ten runs and averaging the number of invalid over valid designs. As discussed earlier, for a similar test, ShipGAN resulted in zero invalid designs; however, approximately 4.32% of designs resulting from GAN were invalid designs. Although this difference is not so significant, it still demonstrates the capability of ShipGAN to produce valid geometries, mainly due to the usage of geometric moments in the SST. Moreover, most invalid designs resulting from GAN have self-intersecting surfaces near the bow of the hull, see Fig. 8.26, which is a local feature. This shows that due to the absence of rich information about the geometry, which in ShipGAN is given with the enhanced SST, the GAN fails to capture the local features of the designs well.

8.4.3 Shape optimisation

The capabilities of such generic parametric modeller can be exploited in different ways to support the designers throughout all three stages of the ship design; i) concept (pre-

Figure 8.26: Examples of invalid (self-intersecting) designs resulted from the GAN model. The red curve indicates the regions of intersection.

liminary) design, ii) contract (full) design, and iii) detail (build) design, especially at the earlier two. As explained earlier, typically, parametric modellers are used solely at the contract phase, where a potential parent design, with resemblance to given design constraints and requirements, selected at the preliminary phase is parameterised and improved with simulation-driven pipelines. Therefore, existing parametric approaches can only handle a particular hull type and cannot aid the designer at the preliminary design stage, where exploring various innovative candidate solutions for the parent design is essential.

With the aid of the generic parametric capabilities of ShipGAN, one can start optimisation from the preliminary design phase with a set of preliminary optimisation criteria, e.g., resistance for an interval of speeds, and constraints, e.g., displacement or volume of displacement, maximum breadth (i.e., to pass through the Panama channel) or maximum draft (i.e., to access specific ports). To showcase this, a simple optimisation setup is formed with optimisation aims to explore the design space, $\mathcal{Z}$, for a container ship with a load-carrying capacity of 3600TEU (Twenty-foot equivalent unit) and an oil tanker with 300,000 tons capacity with improved C_w by solving the optimisation problem in Eq. (8.10) and (8.11).

$$
\begin{aligned}
&\text{Find } \mathbf{z}^* \in \mathbb{R} \quad \text{such that} \\
&\quad C_w(\mathbf{z}^*) = \min_{\mathbf{z}\in\mathcal{Z}} C_w(\mathbf{z}) \\
&\text{subject to} \quad \text{Volume} = 115716.96m^3,\ \text{Over all length} = 242.61m, \\
&\qquad \text{Beam} = 32.2m,\ \text{Depth} = 19.5m. \\
&\qquad 52666.04m^3 \leq \text{Volume of displacement} \leq 58209.84m^3 \\
&\qquad 30.59m \leq \text{Beam at waterline} \leq 33.81m, \\
&\qquad \text{Draft} = 10.8m,\ \text{Length at waterline} = 232.5m.
\end{aligned}
\tag{8.10}
$$

$$
\begin{aligned}
&\text{Find } \mathbf{z}^* \in \mathbb{R} \quad \text{such that} \\
&\quad C_w(\mathbf{z}^*) = \min_{\mathbf{z}\in\mathcal{Z}} C_w(\mathbf{z}) \\
&\text{subject to} \quad \text{Volume} = 470644.41m^3,\ \text{Over all length} = 330.00m, \\
&\qquad \text{Beam} = 58.12m,\ \text{Depth} = 30.00m. \\
&\qquad 296990.90m^3 \leq \text{Volume of displacement} \leq 328253.10m^3 \\
&\qquad 30.59m \leq \text{Beam at waterline} \leq 33.81m, \\
&\qquad \text{Draft} = 20.8m,\ \text{Length at waterline} = 325.50m.
\end{aligned}
\tag{8.11}
$$

The constraints in Eq. (8.10) and (8.11) are set to have a final design similar to existing container ship designs with similar load-carrying capacity, such as the KCS and KVLCC2 hull shown in Fig. 8.4. The KCS is the well-known 3600TEU KRISO container ship designed by the Maritime and Ocean Engineering Research Institute (MOERI). In contrast, the KVLCC2 (KRISO Very Large Crude Carrier) represents a typical 300,000 tons tanker hull form and has been the subject of several experimental and computational studies. The first four constraints are set to have the same load-carrying capacity as KCS and KVLCC2. The last four constraints are physics associated, constraining the submerged part of the hull geometry to ensure C_w is calculated under the same conditions as KCS and KVLCC2.

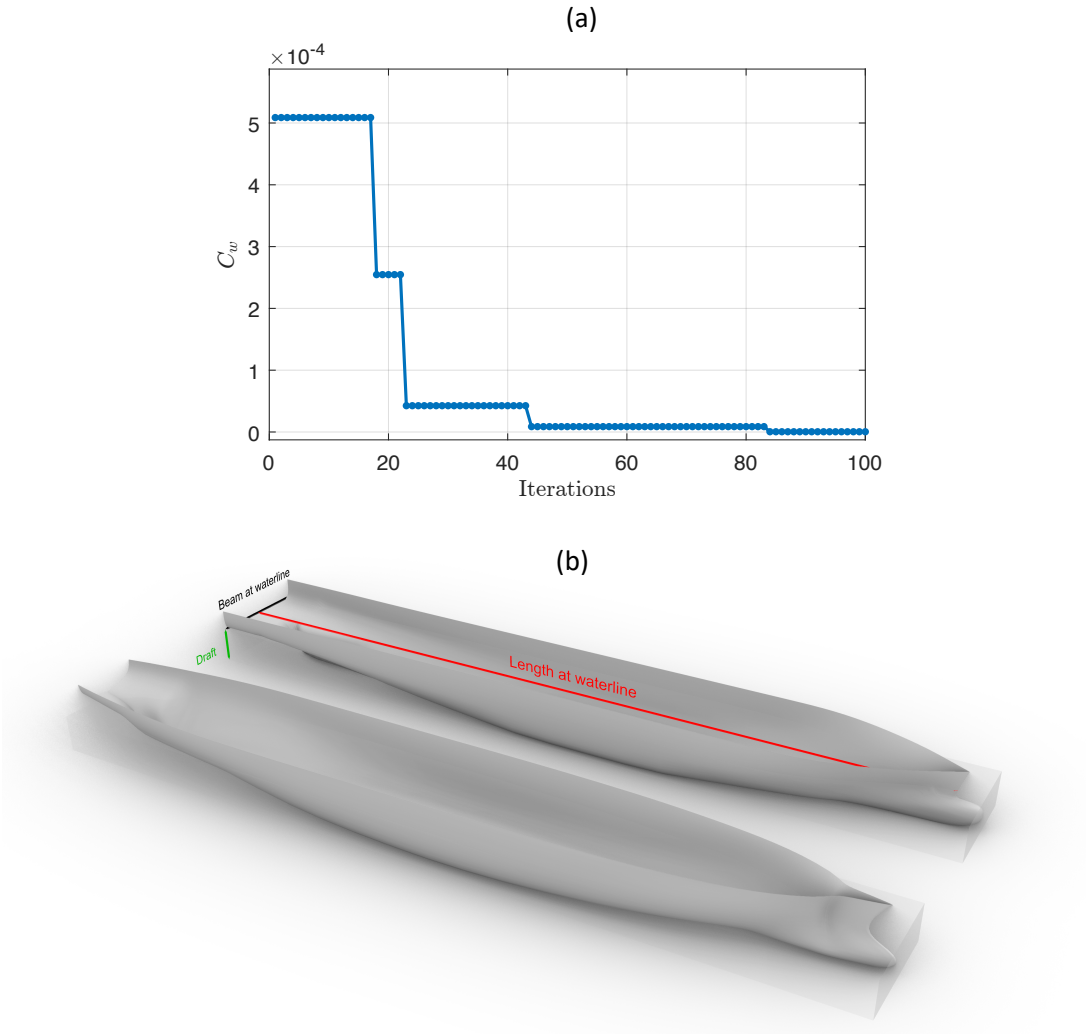


Figure 8.27: (a) Convergence plot of C_w verses first 100 optimisation iterations. (b) 3D surfaces of the KCS hull and the optimised hull having the same particulars as KCS obtained using the ShipGAN model.

The optimisation problems above are solved using Jaya Algorithm (JA), a simple yet efficient optimiser; see more details in [109]. Hydrodynamic simulations for evaluating C_w are performed using a software package based on a linear potential flow theory using Dawson (double-model) linearisation, whose details of the employed formulation, numerical implementations, and validation of the numerical solver are provided in [120]. The computational domain for the free-surface calculation extends from $1Lpp$ upstream to $3Lpp$ downstream and $1.5Lpp$ sideways, whereas Lpp is the length between the perpendicular of a ship hull. A total of $[20 \times 70]$ grid points are used for the free surface, whereas $[25 \times 56]$ grid points are used for the hull discretisation and simulation is performed at Froude number 0.25.

Furthermore, as JA employs a stochastic approach, results may differ in each run; therefore, three different optimisation runs are performed, and the results are averaged in this work. In each run, a total of 500 iterations are performed, and Fig. 8.27(a) and 8.28(a) displays the convergence graph of C_w over the first 100 iterations of the best of five runs. The optimised designs obtained in these cases, along with original KCS and KVLCC2 geometries, are depicted in Fig. 8.27(b) and 8.28(b).

The optimised designs in Fig. 8.27(b) and 8.28(b) satisfy all the design constraints set in Eq. (8.10) and (8.11), with the C_w values of 4.012E-07 and 5.9171E-05, respectively. It is noteworthy that C_w values of KCS and KVLCC2 are 1.846E-03 and 5.972E-03, respectively. So the optimised designs show significant improvement (i.e., reduced C_w value) compared to the existing designs but have the same design characteristics. The improvement in optimised design can also be seen with the reduction of the wave elevation pattern both in terms of transverse and diverging stern waves, which is visible in Fig. 8.29. These results demonstrate the generic parametric capabilities of the ShipGAN modeller that, under different design considerations, it can create not only different design geometries but also with better performance compared to the existing ones.

Furthermore, to test the capability of ShipGAN in the context of conventional parametric modelling, where parametric modellers are developed for a specific hull with a design

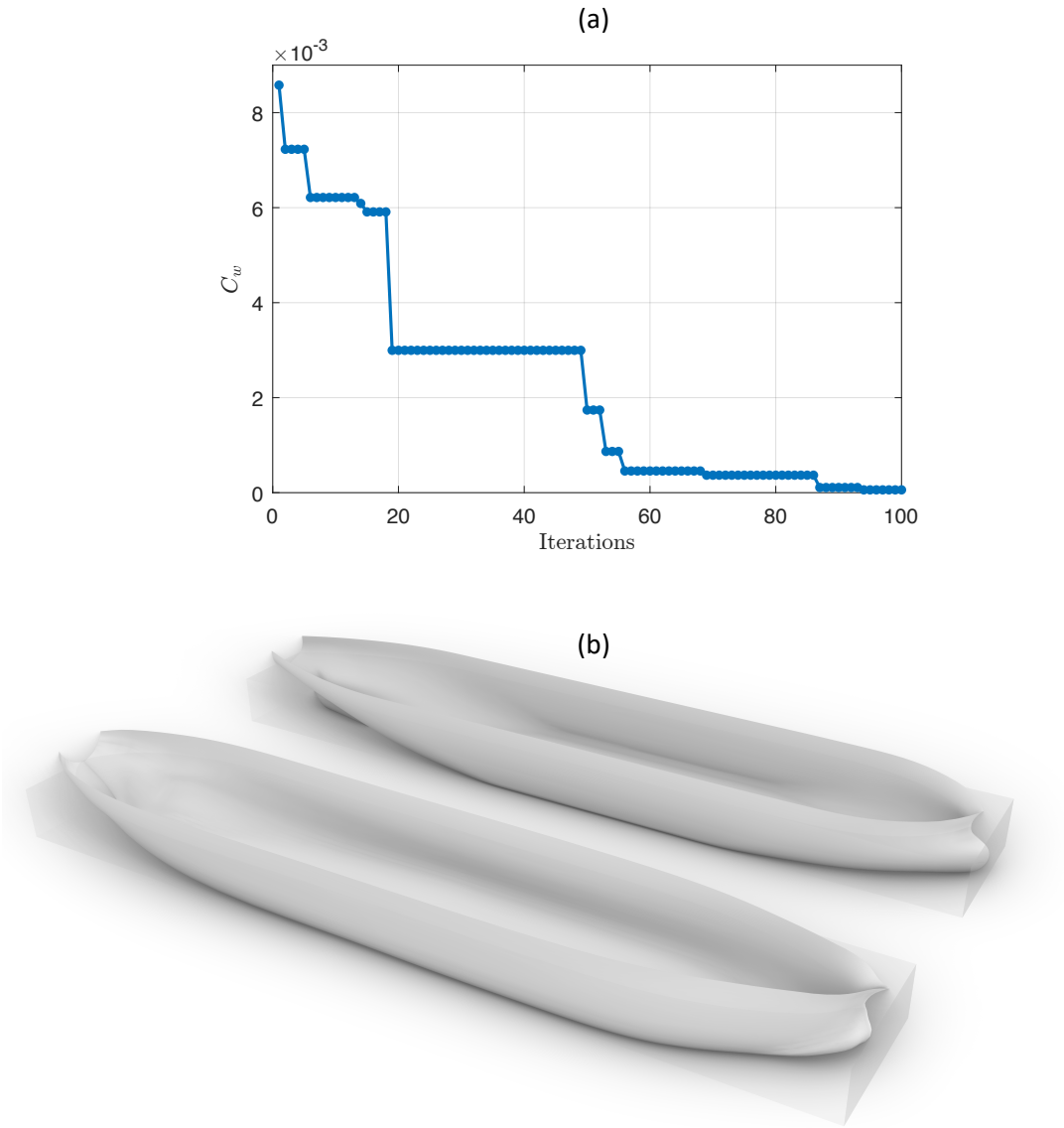


Figure 8.28: (a) Convergence plot of C_w verses first 100 optimisation iterations. (b) 3D surfaces of the KVLCC hull and the optimised hull having the same particulars as KVLCC obtained using the ShipGAN model.

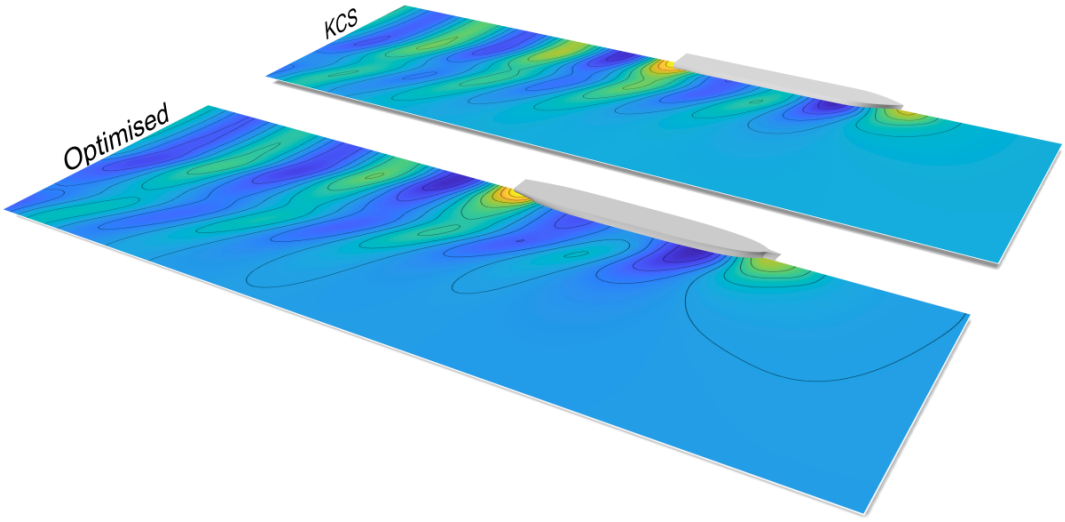


Figure 8.29: Wave pattern of the KCS hull and the optimised hull having the same particulars as KCS obtained using the ShipGAN model.

space capable of creating minor design variations of the parent hull. For this purpose, we first selected the crew supply vessel hull shown in Fig. 8.4 and searched the design space $\mathcal{Z}$ for the closest design $\mathbf{z}_{cs}$. Afterwards, we used $\mathbf{z}_{cs}$ as a parent hull and shrunk the original design space $\mathcal{Z}$ so, as in conventional parametric modelling, it can only produce slight variations of the parent hull. For this purpose, we constructed two design spaces $\mathcal{Z}_{cs_1}$ and $\mathcal{Z}_{cs_2}$ with bounding limits set to $[0.95\mathbf{z}_{cs}, 1.05\mathbf{z}_{cs}]$ and $[0.90\mathbf{z}_{cs}, 1.10\mathbf{z}_{cs}]$, respectively, allowing 5% and 10% variations of $\mathbf{z}_{cs}$. The optimisation is performed as follows:

$$
\begin{aligned}
&\text{Find } \mathbf{z}^*_{cs} \in \mathbb{R} \quad \text{such that} \\
&\quad C_w(\mathbf{z}^*_{cs}) = \min_{\mathbf{z}_{cs} \in \mathcal{Z}_{cs_1}/\mathcal{Z}_{cs_2}} C_w(\mathbf{z}_{cs}) \\
&\text{subject to} \quad 53.96m^3 \leq \text{volume of displacement} \leq 59.64m^3, \\
&\qquad\qquad\quad 5.55m \leq \text{Beam at waterline} \leq 6.13m, \\
&\qquad\qquad\quad \text{Draft} = 0.9m, \ \text{Length at waterline} = 34.69m.
\end{aligned}
\tag{8.12}
$$

Note that the first four constraints in Eq. 8.10 and 8.11 are not included in Eq. 8.12

as the design spaces $\mathcal{Z}_{cs_1}$ and $\mathcal{Z}_{cs_2}$ intrinsically satisfy these constraints. The results of this experiment are shown in Fig. 8.30. The C_w values of $\mathbf{z}_{cs}$ and its optimised versions resulting from $\mathcal{Z}_{cs_1}$ and $\mathcal{Z}_{cs_2}$ are $3.649E-03$, $2.090E-03$ and $8.677E-07$, respectively. It can be seen that the optimised designs have a significant reduction in their c_w values, especially design resulting from the $\mathcal{Z}_{cs_2}$ as it provides higher design variations compared to $\mathcal{Z}_{cs_1}$.

8.5 Conclusions and future works

In this chapter, we demonstrated the first application of deep convolutional generative adversarial networks for the parametric modelling of ship hulls. We first present a new architecture of GANs by introducing a space-filling layer to ensure the generator can cover all design classes. We also inputted GMs to the network along with the shape representation in the form of SST. GMs provide rich information about the overall design's geometric stricture, and for the ship design, they also induce the notion of physics. This approach results in the trained generator providing 100% geometrically valid geometries and practically feasible shapes.

In this chapter, we have mainly focused on developing a generic parametric modeller. In the future, our immediate goal is to explore the capabilities of such modellers at the preliminary design stage and how a more practical optimisation problem can be set up for a user to exploit generic parameterisation abilities fully. In the long term, we also aim to train GANs simultaneously for physics (similar to reduce-order modelling) and geometries with a fully connected layer for physics prediction.

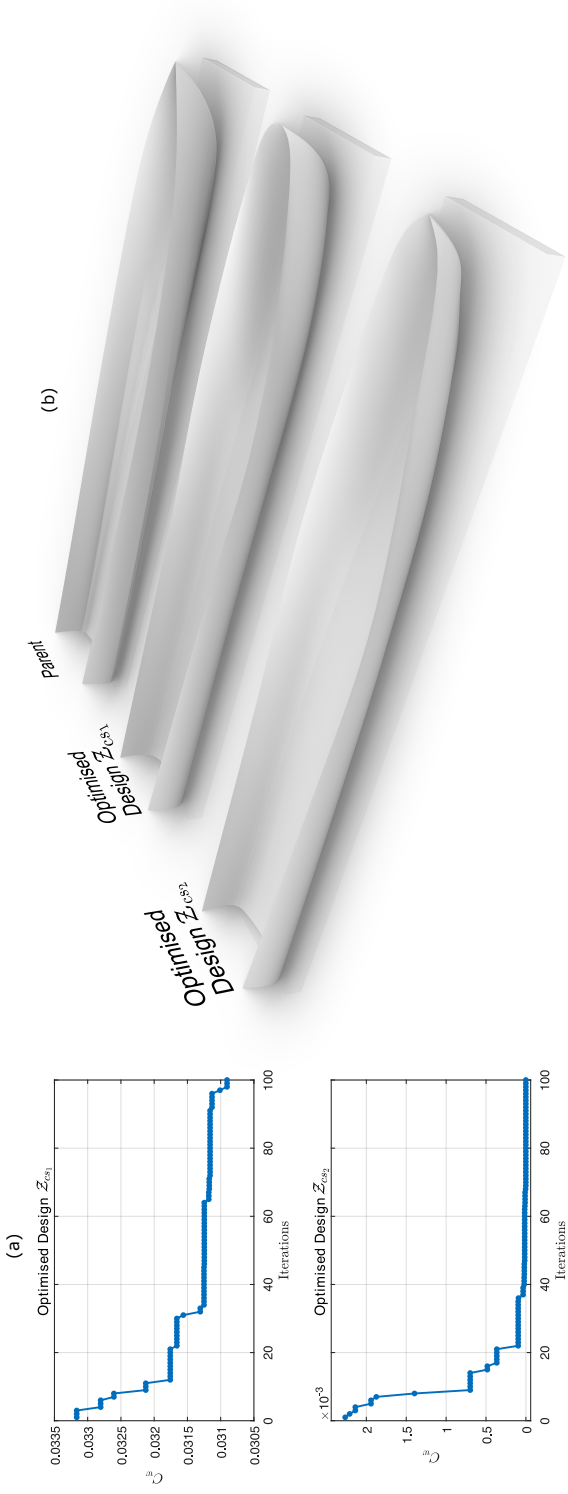


Figure 8.30: (a) Convergence plots of C_w verses first 100 optimisation iterations performed in $\mathcal{Z}_{cs_1}$ and $\mathcal{Z}_{cs_2}$. (b) 3D surfaces of $\mathbf{z}_{cs}$ and its optimised variants resulting from $\mathcal{Z}_{cs_1}$ and $\mathcal{Z}_{cs_2}$.

Chapter 9

GenYacht: An interactive generative design system for yacht Hull design

9.1 Introduction

In this chapter, we aim to take the next step in the computer-aided preliminary yacht hull design by interactively inducing the user preference on designs into the design of space exploration. This is achieved by introducing a new interactive design system, *GenYacht*, which brings the benefit of the interactive and generative design to the preliminary design stage to generate user-driven hull forms with better performance. However, the proposed interactive technique can also be utilised for design applications in maritime and other engineering fields.

Generative design is an algorithm-driven design process to empower experienced or novice designers to generate the desired number of optimum alternatives for an initial design. Instead of a single solution, the generative design creates potentially various solutions satisfying the given design requirements. It facilitates the designer with the comfort

of selecting a solution that best meets their needs [35]. Even for the most experienced designers, their intuition might be limited when manually exploring an unprecedentedly large design space. In generative design, a basic layout of an input CAD model is first created. Design specifications and constraints are then defined. Various computational simulations are later executed to obtain a set of optimised solutions [13].

Interactive design is a process in which a given design space is explored, and a target design is evaluated based on human subjective evaluation. The interaction with a human evaluator facilitates the generation of a solution that incorporates human intuition without explicitly codifying them into the design process. In interactive design systems, the user carries out the design exploration either with interactive interfaces [9, 158, 182, 183] or by integrating the meta-heuristics with the interactive interfaces to semi-automate the exploration process [45,184–186]. In the latter approaches, users are interactively involved at each iteration/generation of an optimiser and guide the optimisation process towards the promising regions of the design space. In this approach, an initial population is first created consisting of randomly sampled designs. A user then performs interaction for selecting a design [187], or they can rate all the designs shown [188]. The optimiser then performs an iteration to generate designs similar to the selected or highly-rated design(s). The creation of similar designs is usually done utilising a distance-based metric [184,186]. This iterative and interactive process continues until the user reaches a preferred or satisfactory design.

As designs generated in each iteration are based on the user's selection(s) in the previous iteration, starting the interactive process with the randomly generated designs, which are mostly clustered and non-uniformly distributed, can restrain the user from exploring all the design possibilities. Furthermore, distance-based exploration can force the optimiser to converge to similar designs at a fast rate; therefore, a large portion of design space can be left unexplored, which will be proven via experimentations later in this work.

The proposed system is based on novel interactive and generative design techniques, which run in parallel during the hull form creation. The generative design technique (GDT) provides a promising way to explore the design space and generate well-diverse

design alternatives automatically. A design space is first created based on the upper and lower bounds of the geometric parameters of the parent hull. GDT generates a set of N design alternatives in this space. These alternatives are uniformly distributed in the design space, and each design represents a particular location in the design space (see the output of GDT in Fig. 9.1, which illustrates the hull forms generated in the two-dimensional design space). The interactive design involves user preference/intuition interactively during the design process, thereby guiding the design exploration towards a more promising region of design space. At this step, hull designs are searched with GDT, and three-dimensional (3D) surface models for the yacht hulls are generated using Khan et al.'s [4] parametric design technique. Afterwards, these models are presented to the user with their physical properties, such as hydrostatics and resistance. The user then interacts while selecting a design(s), and the design space is refined based on the chosen design(s). An overall workflow of GenYacht is shown in Fig. 9.1. In this work, the refinement of the design space is done using a novel space-shrinking technique (SST), which shrinks the design space and generates new designs in the shrunk space for the next interaction. The interactive process continues until the user reaches a hull design with desired characteristics. It is noteworthy that the user selections are made not only based on the hull's performance but also according to its form appearance.

9.2 Related works

Triggered by digital and manufacturing advances, interactive and generative design has received significant attention in computer-aided design (CAD) and computer graphics communities. We mainly focus our literature review on interactive and generative design for exploring design space for parametric CAD shapes. In this section, we first review prior works in interactive design, followed by a discussion of existing studies in naval architecture and a brief introduction to generative design systems.

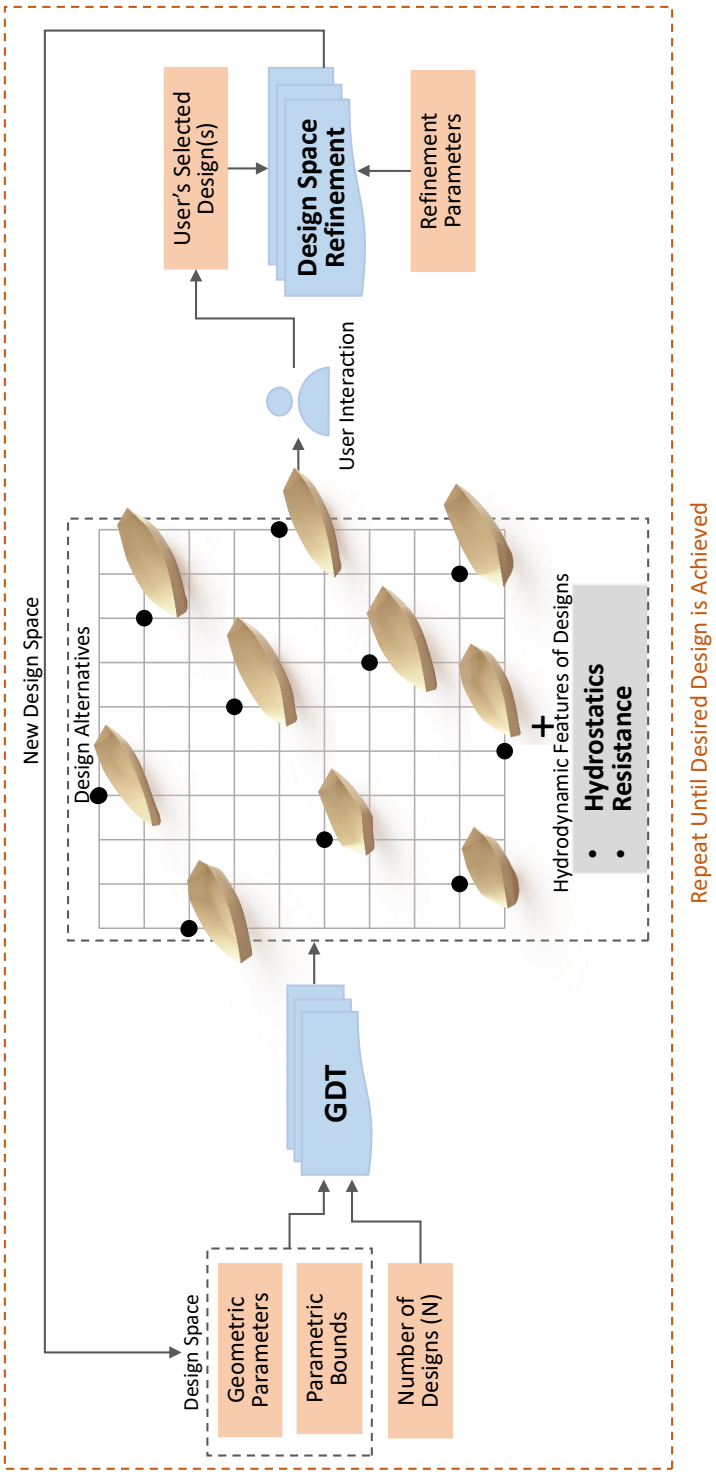


Figure 9.1: Overall workflow of GenYacht. A design space, formed with geometric parameters and their limits, is inputted into the generative design technique (GDT), which generates N uniformly distributed hull design alternatives. Among them, a user selects a design based on its appearance and physical properties such as hydrostatics and resistance. The design space is then refined according to the selected designs, which is again fed into GDT for creating new designs. This process is repeated until the final design(s) are achieved.

9.2.1 Interactive design

Computational design tools help users to create digital designs for various applications, which is done using optimisation techniques, interactive interfaces or a combination of both to create hybrid systems. These tools guide users into exploring a given parametric design space for specific physical criteria. Mainly, interactive interfaces (commonly used in the computer graphics community) are developed for particular design applications, which are used to synthesise and assemble components to explore design variations. For instance, Bole [182] developed a transformation tool to interactively manipulate geometric parameters for a ship hull design. Interactive tools have also been proposed for synthesising three-dimensional (3D) characters [189], procedural modelling of the architectural structures [190] and for 3D modelling of garment patterns [191]. Some researchers have also developed interactive techniques for exploring 3D shape variations [192,193] and for prediction of their physical properties, such as the aerodynamics of automotive [158], and mechanical stress of 3D components [9].

Interactive design approaches have also been coupled with meta-heuristics, which usually refer as Interactive Evolutionary Computation (IEC). In IEC, human evaluation is used as a component of objective function during solution space exploration for an optimum solution. During exploration, the user's intuitive assessment of a solution is incorporated to create a user-oriented or user-centred design. The incorporation is carried out differently during optimisation for different end objectives. In IEC, an overwhelming majority of works proposed interactive genetic algorithms (IGA) for various design applications. IGAs are based on the typical genetic algorithm (GA) principle. Brintrup et al. [184] proposed an IGA to incorporate the qualitative and quantitative criteria for ergonomic chair design. In their technique, a user plays the role of the qualitative criterion guiding the optimisation to the desired location in the design space. First, an initial population of solutions are presented to the user, where they rate the designs on a Likert scale, and these ratings act as a fitness value for each design. Therefore, the designs with higher scores become parents,

and the evolutionary process is carried out to generate designs similar to the parent.

A multi-stage IGA (MS-IGA) was proposed by Dou et al. [188]. At the initial stage, MS-IGA generates populations of simple designs and as the interactive evolutionary process continues the design becomes complex. Dou et al. argue that this helps to minimise user fatigue during an interaction, *which is one of the significant drawbacks of the IGA-based systems* [194]. In IEC, user fatigue is the inability of the user to select potential designs during the design interaction due to physical or psychological exhaustion [184]. In another study, along with the user rating, Dou et al. [195] incorporated the time spent to evaluate each design to calculate its fitness value. The incorporation of evaluation time simulates user hesitancy in the design process. The performance of their works [188, 195] was validated with a car dashboard design.

Poirson et al. [186] elicited user perception about the product design using IGA. In their approach, a user first selects the design which mainly represents a given semantic attribute. Then, GA moves the population of solutions closer to the chosen design via a distance function. Poirson et al. also performed different experiments on the parameter tuning of GA, as the convergence of GA mainly depends on these tuning parameters. Hernandez et al. [196] addressed the problem of Unequal Area Facility Layout using an IGA. However, instead of presenting the entire population of designs, Hernandez et al. [196], and Machwe and Parmee [197] utilised clustering techniques to enable a user to evaluate the representative design of each cluster, thereby ameliorating user fatigue. Gu et al. [198] incorporated the Neural Network-based learning technique, General Regression Neural Network (GRNN), into IEC to approximate the user's aesthetic perception during the interactive evolutionary process. IGA-based systems have also been proposed for aircraft [199], software design [200] and structural design [201].

The literature also contains some recent techniques [202] to interactively prune the Pareto Front solution set at each generation of multi-objective GA, which helps to reduce the size of the Pareto front and to obtain the desired Pareto optimal solutions at the end of the evolutionary process. Recently, a few researchers have diverted also their attention

to utilising other meta-heuristic algorithms such as Particle Swarm [187] and Teaching-Learning-Based Optimisation [185] to develop IEC-based design systems.

IGA has been used for various design applications; however, to the best of our knowledge, Duchateau's work [45] is the only example related to the subject of the present study in the field of naval architecture. In [45], Duchateau proposed an IGA-based technique to allow users to create and select designs based on insight gained during the design exploration process. The proposed method was applied for the preliminary design of a mine-countermeasures vessel. Duchateau argued that the complexity of the ship design hinders designers from exploring vast and potentially more regions of the design space with traditional design techniques. Therefore, an interactive and evolutionary approach was proposed to steer the optimiser to explore more promising design solutions gradually. In naval architecture, DeNucci's work [203] is another example of the user's involvement in the design process, which focuses on capturing and integrating the design rationale (i.e., the reasoning behind the design decision) at the conceptual stage of ship design. DeNucci developed a Rationale Capture Tool (RCT) to incorporate design rationale as user experience and performance into the design process, thereby linking the users' design configuration preferences to the ship's performance at its preliminary design phase.

In this work, we propose a new interactive design to overcome the drawbacks above of the IGA-based system. Therefore, in Table 9.1, we describe some advantages of GenYacht over typical IGA-based design systems.

9.2.2 Generative design

During the last few years, generative design techniques have played a critical role in automating the exploration of parametric design spaces. Unlike traditional optimisation-based design exploration, GDTs explore large design spaces to find a *variety* of optimal alternatives that give users the ability to choose a design that best fits their needs. The literature contains many efforts from researchers in design exploration techniques for the preliminary design of naval vessels [47, 49, 50, 204]. However, these techniques are not de-

Table 9.1: Comparison between IGA-based systems and GenYacht

No.	IGA	GenYacht
1	Interactive process starts with the initial population of randomly generated designs, which, in most cases, are not well spread out in the design space. Therefore, this can limit users from well exploring all regions of design space [193].	The interactive process starts with uniformly distributed designs, covering all the design possibilities within the design space. This allows users to effectively explore the entire design space.
2	Requires tuning of optimisation parameters, such as selection operator, crossover and mutation probability, for desirable results, which is non-trivial for most of the users [13].	Does not require parameter tuning of the optimisation-specific parameters. The only user-defined parameter is the shrink/expand rate, which controls the diversity of designs in each interaction. According to the experiments conducted in this work, the shrink/expand rate does not affect the optimiser's performance to generate uniformly distributed and non-collapsing designs.
3	Selection of suitable distance metric(s) is critical [186] to converge (i.e., get similar) the initial population of solutions towards the user-selected designs.	The design space is shrunk in each interaction while eliminating the non-preferred regions, which aids the optimiser in converging to the user-preferred designs.
4	It is hard to maintain the high variations between designs in the interactions.	A user can create significantly diverse designs at each interaction using space-filling and non-collapsing criteria (will be discussed in Section 9.3).
5	Starting the interactive process with random designs requires users to carry out many interactions to explore all the design possibilities. Therefore, this higher number of design evaluations can increase user fatigue [184, 197], thereby converging towards the local optimal and undesirable solutions.	Starting the interactive process with uniformly distributed design can help users explore more design possibilities with fewer design evaluations, reducing user fatigue. Moreover, the space-shrinking technique better controls the total number of interactions performed.

veloped in the context of generative design. Therefore, we can only explore a limited region of design space to generate single or Pareto designs, which are usually a *slight variation* of the parent shape. To give some background to the readers from naval architecture, here, we mentioned some existing *generative design systems developed for parametric design exploration* and their limitations.

A random search-based generative system, *Genoform*, was developed by Krish et al. [35] for parametric design exploration, in which variation between designs is achieved via the Euclidean distance-based similarity criterion. Genoform cannot explore a design space well due to its random search nature, proven via experimentation in [13]. An iterative design exploration system, *Fractal* [205], was developed by Autodesk, which provides n^I design possibility for a given parametric shape. Here, n represents the geometric parameters, and I is the number of levels for each parametric range. Another system called *Dream Lens* was proposed by Matejka et al. [206] to explore and visualise many generatively created designs. Dream Lens explores performance spaces for the given problem domains. Recently, Khan et al. [8] proposed a Psycho-physical distance metric to induce human perception into the design process for the exploration of diverse shapes.

Similarly, Kazi et al. [207] developed *DreamSketch*, a generative design platform for exploring design sketches at the conceptual stage. The usability of this system requires users to have digital sketching skills. Moreover, Zaman et al. [208] devised *GEM-NI*, a generative design software for exploring two-dimensional shapes. Later, an extension of GEM-NI called *MACE* was also proposed by Zaman et al. [209] to enhance the capability of visualisation of design alternatives. Gunpinar and Gunpinar [210] proposed a generative design approach based on a particle tracing algorithm, and recently, Khan and Awan [13] developed a generative design system, *DesignN*, for exploration of CAD shapes with continuous and discrete parameters. However, in [210] and [13], no physical performance criterion was evaluated during the design exploration.

Some researchers have also introduced some application-specific generative design systems, such as *ParaGen*, *Dexen* and *GENE_ARCH*, which were introduced by Turrin et

al. [7], Patrick [211] and Caldas [212] for exploring parametric structures, façade and energy efficient building designs, respectively.

9.3 Method overview

In this section, the algorithmic details of GenYacht will be introduced. After describing the basic terminology and the generative design approach, the proposed interactive design approach will be presented in line with the space-shrinking technique and GenYacht's user interface.

9.3.1 Basic terminology and generative design techniques (GDT)

Let a design space $\mathcal{X}$ formed for a parent yacht hull m, which is represented using a set of geometric parameters, $\mathbf{x}_m = \{x_{m,k}, k = 1, 2, \ldots, n\} \in \mathcal{X} \subseteq \mathbb{R}^n$. $\mathcal{X}$ is a subset of $\mathbb{R}^n$ and is bounded by the lower $\mathbf{x_m^l}$ and upper $\mathbf{x_m^u}$ bounds of geometric parameters (i.e. $\mathcal{X} := \{x^l_{m,k} \leq x_{m,k} \leq x^u_{m,k}, \forall k \in \{1, 2, \ldots n\}\}$).

It is impractical, if not impossible, for a user to manually iterate through all the astronomical possibilities of hull designs in $\mathcal{X}$. Therefore, our objective is to explore $\mathcal{X}$ with the aid of an optimiser to find a set $\mathcal{N}$ consisting of N uniformly distributed diverse hull forms ($\mathcal{N} = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \ldots, \mathbf{x}_N\} \in \mathcal{X}$). Here, N is a user-defined parameter, and each design in $\mathcal{N}$ represents a specific location in $\mathcal{X}$. To obtain the set $\mathcal{N}$, Khan and Gunpinar's approach [38] is adopted, which is briefly explained in this subsection. This approach utilises Audze and Eglais [213] space-filling criterion ($F_1(\mathcal{N})$) to find uniformly distributed designs (see Equation 9.1).

$$F_1(\mathcal{N}) = \sum_{p=1}^{N-1} \sum_{q=p+1}^{N} \frac{1}{\mathcal{D}(\mathbf{x}_p, \mathbf{x}_q)^2} \tag{9.1}$$

where

$$\mathcal{D}(\mathbf{x}_p, \mathbf{x}_q) = \sqrt{\sum_{k=1}^{n} (x_{p,k} - x_{q,k})^2} \tag{9.2}$$

Here, $\mathcal{D}(\mathbf{x}_p, \mathbf{x}_q)$ is the Euclidean distance between the designs p and q. Minimisation of $F_1(\mathcal{N})$ favours the uniform distribution of the N designs in $\mathcal{X}$.

In the case of high-dimensional design spaces, the space-filling criterion favours the placement of designs to the design space's boundaries, which is undesirable. Therefore, the space-filling designs are searched within the class of Latin-hypercube with a criterion of non-collapsingness between designs. This criterion divides each dimension of $\mathcal{X}$ into N intervals and ensures that the two designs do not share the same range. It is incorporated into the search process using Equation 9.3, which calculates the number of intervals that N designs share. Minimising this equation creates complete or semi-non-collapsing designs depending on a user-controlled parameter Ω, which adjusts the weight for $F_2(\mathcal{N})$.

$$F_2(\mathcal{N}) = \Omega \times \sum_{p=1}^{N-1} \sum_{q=p+1}^{N} \mathcal{K}(\mathbf{y}_p, \mathbf{y}_q) \tag{9.3}$$

$$\mathcal{K}(\mathbf{y}_p, \mathbf{y}_q) = \sum_{j=1}^{n} f(y_{p,k}, y_{q,k}) \tag{9.4}$$

$$f(y_{p,k}, y_{q,k}) = \begin{cases} 1 & \text{if } y_{p,k} = y_{q,k} \\ 0 & \text{otherwise} \end{cases} \tag{9.5}$$

In Equation 9.3, $\mathcal{K}(\mathbf{y}_p, \mathbf{y}_q)$ denotes the number of intervals that the designs p and q share, and $\mathbf{y}_p$ and $\mathbf{y}_q$ are the discrete representations for $\mathbf{x}_p$ and $\mathbf{x}_q$, respectively. To calculate the discrete value $(y_{p,k})$ of k^{th} geometric parameter $(x_{i,k})$ for the i^{th} design, its range between lower $(x_{i,l}^k)$ and upper $(x_{i,u}^k)$ bounds is first partitioned into N intervals $[x_{i,k}^l = x_{i,k}^1, x_{i,k}^2, \ldots, x_{i,k}^N = x_{p,k}^u]$ and an integer coordinate r is then assigned to y_i^k as follows: $\forall r \in \{1, 2, \ldots, N\}$, $(x_{i,k}^r \leq x_{i,k} < x_{i,k}^{r+1}) \Rightarrow (y_{i,k} = r)$.

During the design exploration for the N designs, the cost function $\mathbf{F}(\mathcal{N})$ in Equation

9.6 is *minimised*.

$$\mathbf{F}(\mathcal{N}) = \sum_{p=1}^{N-1} \sum_{q=p+1}^{N} \frac{1}{\mathcal{D}(\mathbf{x}_p, \mathbf{x}_q)^2} \quad + \quad \Omega \times \sum_{p=1}^{N-1} \sum_{q=p+1}^{N} \mathcal{K}(\mathbf{y}_p, \mathbf{y}_q) \tag{9.6}$$

In this approach, the design exploration process starts by generating an initial population ($\mathbf{P}$) consisting of N sub-populations ($\mathbf{P} = \{p_L, L = 1, 2, \ldots N\}$). The L^{th} sub-population of $\mathbf{P}$ consists of s randomly sampled designs ($p_L = \{\mathbf{x}_g, g = 1, 2, \ldots s\}$) in $\mathcal{X}$. For each solution, $\mathbf{P}$ contains a sub-population of size s. During the convergence, an optimiser guides all the sub-populations to their optimum position under the consideration of each subpopulation's best solution (i.e. a solution that minimises $\mathbf{F}(\mathcal{N})$). Initial solution set $\mathcal{N} = \{\mathbf{x}_{p_1}, \mathbf{x}_{p_2}, \ldots, \mathbf{x}_{p_N}\}$ is first obtained from $\mathbf{P}$ containing N solution; one solution from each sub-population using a greedy-selection strategy [38]. The initial $\mathcal{N}$ contains the combination of solutions which gives a minimum value of $\mathbf{F}(\mathcal{N})$. During the optimisation, each iteration is completed by performing N sub-iterations, and a sub-iteration is completed after updating all the designs in a sub-population using an optimiser. After the convergence, GDT returns an optimal set $\mathcal{N}_{op}$ of N space-filling designs. Algorithm 5 summarises the stepwise procedure of GDT.

Different optimisers, such as Genetic Algorithm (GA) [214], Particle Swarm Optimisation (PSO) [215], Artificial Bee Colony (ABC) [216], Teaching-Learning-Based Optimisation (TLBO) [217] and Jaya Algorithm (JA) [218], have been integrated into GDT and a final selection for GenYacht system was made based on optimisers' performance and computational complexities. The results of these optimisers will be shown in Section 9.4.1. Fig. 9.2 (a) and (b) show the randomly distributed designs and uniformly distributed designs created using GDT. It can be seen that designs generated using GDT are well distributed and cover all the regions of the design space.

Algorithm 5 The pseudo-code of generative design algorithm

function $GDT(\mathcal{X}, N, s, \Omega)$
Input: Create a parent hull m and parametrise it with n geometric parameters $(x_{m,1}, x_{m,2}, \ldots, x_{m,n})$.
Input: Initialise number of designs to be created (N), sub-population size (s) and parameter Ω.
Input: Define the design space with lower and upper bounds of n parameters, $\mathcal{X} := \{x^l_{m,k} \leq x_{m,k} \leq x^u_{m,k} \forall k \in \{1, 2, \ldots n\}\}$.
Randomly create an initial population $(\mathbf{P})$ consisting of N sub-populations $(p_1, p_2 \ldots, p_N)$ of size s.
Select N initial best designs $(\mathcal{N} = \{\mathbf{x}_{p_1}, \mathbf{x}_{p_2}, \ldots, \mathbf{x}_{p_N}\})$ one from each sub-population.

while termination criterion is not satisfied **do**
  for $L = 1$ to N **do**
    for $g = 1$ to s **do**
      Update design $\mathbf{x}_g$ of p_L using an meta-heuristic optimiser and obtain updated design $\mathbf{x}'_g$.
      Calculate cost value $\mathbf{F}(\mathcal{N}')$ and $\mathbf{F}(\mathcal{N})$ for $\mathcal{N}' = \{\mathbf{x}'_g, \mathbf{x}_{p_2}, \ldots, \mathbf{x}_{p_N}\}$ and $\mathcal{N} = \{\mathbf{x}_g, \mathbf{x}_{p_2}, \ldots, \mathbf{x}_{p_N}\}$.
      if $\mathbf{F}(\mathcal{N}') < \mathbf{F}(\mathcal{N})$ **then**
        Replace the old design $\mathbf{x}_g$ with $\mathbf{x}'_g$ in p_L
      else
        Reject the new design $\mathbf{x}'_g$ and keep $\mathbf{x}_g$ in p_L
      end if
    end for
    Obtain the updated p_L and set as p'_L.
    Find the new best design $\mathbf{x}'_{p_L}$ from p'_L.
    Replace $\mathbf{x}_{p_L}$ with new $\mathbf{x}'_{p_L}$ in set $\mathcal{N}$ (i.e. $\mathcal{N} = \{\mathbf{x}'_{p_1}, \mathbf{x}_{p_2}, \ldots, \mathbf{x}_{p_N}\}$).
  end for
end while
return Optimal design set $\mathcal{N}_{op}$.

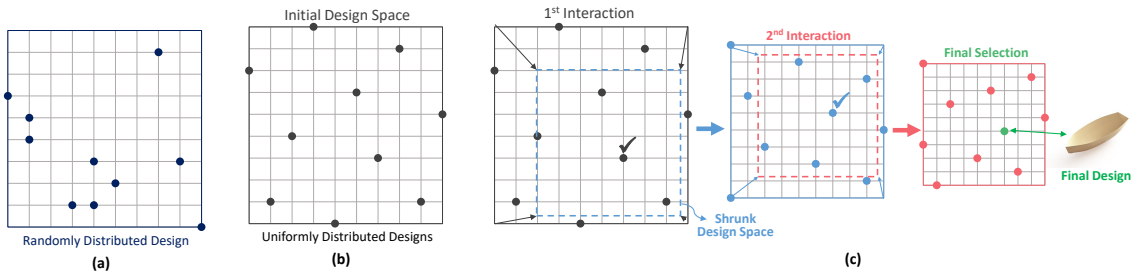


Figure 9.2: Illustration of designs generated via random sampling in a two-dimensional design space (a). While the designs generated using GDT in the same two-dimensional space are uniformly distributed because of space-filling and non-collapsing criteria (b). The interaction process started with GDT-generated designs, and in each interaction, the design space is shrunk towards the user selection (c). The selection of design is indicated with a tick mark.

Constrained design spaces

GenYacht also allows users to explore constrained spaces composed of feasible (i.e. designs satisfying the constraints) and infeasible (i.e. designs violating the constraints) designs. GDT should only generate feasible designs. Therefore, in this work, Deb's heuristic constrained handling method [219] was utilised, which uses a tournament selection operator. This operator selects two designs and compares them with each other. A design p is said to be constrained-dominate other design q if any of the following heuristic rules are true:

1. The design p is feasible and design q is not.
2. The designs p and q both are infeasible, but design p violate less number of constraints.
3. The designs p and q both are feasible, but design p has minimum cost function value.

The design p is selected only if constrained-dominate design q. If both designs, p and q, are infeasible and have the same number of constraint violations, the design with better cost value is then selected.

9.3.2 Interactive design approach

In an interactive design stage, N hull designs generated via GDT are shown to the user along with their physical properties such as form coefficients, residuary and frictional resistance, metacentric radius, metacentre, longitudinal and transverse moments of inertia, longitudinal and vertical centre of buoyancy and flotation. The user then selects the designs according to the hulls' overall appearance and physical properties. This interaction step allows users to compare designs based on their requirements and helps them make an appropriate design decision. Once the desired hull form is selected, the design space is refined based on the selected design. The refined design space is then imported into GDT to generate new designs in the following interaction step. This interaction procedure is repeated multiple times until a desirable number of designs is obtained. Fig. 9.2 (c) illustrates the implementation of the proposed interactive design approach on a two-dimensional design space. As shown in this Figure, the design space formed in the previous interaction shrinks at each interaction by focusing on the preferred designs. This way, the user's interest region can be better scanned.

Our design space shrinking process follows the analogy of *woodcarving* in which a carver first selects a large piece of timber (usually bigger than the size of the final form) to create the desired artefact. They then remove the large chunks of wood to achieve a general shape. Afterwards, the carver scrapes the pieces of timber step-by-step and gradually proceeds to a final shape. Such material scraping can be reflected as an exponential decay. In the initial interactions, the design space shrinks at a faster rate, and it decreases exponentially as the interaction process continues. The algorithmic details of the proposed SST are given in the next section.

Space-shrinking technique (SST)

In the proposed interactive approach, an initial design set $\mathcal{N}_{op}$ is first generated using GDT and an interaction loop between the user and GenYacht is then completed involving three

steps: (2) the user selects preferable design(s) among the ones generated by GDT, (3) the design space is refined based on the selection(s), and (3) the shrunk space is inputted to GDT for the creation of new designs)for the next interaction. At the end of the multiple interaction loops, single or multiple preferred designs are obtained.

For the sake of simplicity, a single hull selection will be considered in the method's explanations. In the T^{th} interaction loop, the user selects a t^{th} design ($\mathbf{x}_t$) from $\mathcal{N}_{op}^{T-1} = [\mathbf{x}_{p_1}, \mathbf{x}_{p_2}, \ldots, \mathbf{x}_{p_N}]$, which is obtained from GDT in the $(T-1)^{th}$ (T is integer) interaction. After the user selection, a new design space, $\mathcal{X}^T$, is formed while shrinking the previous design space ($\mathcal{X}^{T-1}$) based on the selected/preferred design $\mathbf{x}_t = (x_{t,1}, x_{t,2}, \ldots, x_{t,n})$, and a new design set $\mathcal{N}_{op}^T$ is obtained from $\mathcal{X}^T$. The shrinking of design space is performed by calculating new lower ($\acute{\mathbf{x}}^{\mathbf{l}}_{\mathbf{m}}$) and upper ($\acute{\mathbf{x}}^{\mathbf{u}}_{\mathbf{m}}$) bounds using Equation 9.7 for $\mathcal{X}^T := \{\acute{x}^l_{m,k} \leq x_{t,k} \leq \acute{x}^u_{m,k}, \forall k \in \{1, 2, \ldots n\}\}$.

$$\begin{cases} \acute{x}^l_{m,k} = x^l_{m,k} + \left|\lambda^T_- \times \mathcal{R}^l_T\right| \\ \acute{x}^u_{m,k} = x^u_{m,k} - \left|\lambda^T_- \times \mathcal{R}^u_T\right| \end{cases} \quad \text{where} \quad k \in \{1, 2, \ldots, n\} \tag{9.7}$$

Here, λ^T_- is the shrink rate initialised by the user in the T^{th} interaction and ranges between $0 < \lambda^T_- \leq 1$. When λ^T_- is zero, the space-shrinking process terminates. $\mathcal{R}^l_T$ and $\mathcal{R}^u_T$ are the continuous growth and decay parameters, respectively, which are computed using Equation 9.8 after the interaction T.

$$\begin{cases} x_{t,k} = x^l_{m,k} \times \exp(\mathcal{R}^l_T \times T) \\ x_{t,k} = x^u_{m,k} \times \exp(\mathcal{R}^u_T \times T) \end{cases} \tag{9.8a}$$

Equation 9.8a represents the continuous exponential decay of the design space during the interactive process. $x_{t,k}$ represents the amount after shrinkage and $x^l_{m,k}$ is the initial amount at the T^{th} interaction. Solving the Equation 9.8a for R^l_T and R^u_T yields Equation 9.8b.

$$\begin{cases} \mathcal{R}_T^l = \ln\left(\frac{x_{t,k}}{x_{m,k}^l}\right) \times \frac{1}{T} \\ \mathcal{R}_T^u = \ln\left(\frac{x_{t,k}}{x_{m,k}^u}\right) \times \frac{1}{T} \end{cases} \tag{9.8b}$$

After obtaining the shrunk space, $x_{m,k}^l$ and $x_{m,k}^u$ are set equal to $\acute{x}_{m,k}^l$ and $\acute{x}_{m,k}^u$ respectively. Additionally, GenYacht can expand the design space. If the user is not satisfied with the designs in an interaction he/she can expand the design space instead of shrinking the space. During the expansion, the upper and lower bounds of $\mathcal{X}^T$ are constrained by the upper and lower bounds of the initial design space $\mathcal{X}$ (i.e., $\mathcal{X}^T := \{\acute{x}_{t,k}^l \leq x_{t,k} \leq \acute{x}_{t,k}^u : (\acute{x}_{t,k}^u \leq x_{m,k}^u) \wedge (\acute{x}_{t,k}^l \geq x_{m,k}^l) \forall k \in \{1, 2, \dots n\}\}$), which limits the new design space from over-expanding the initial design space. The design space is expanded using Equation 9.9.

$$\begin{cases} \acute{x}_{m,k}^l = x_{m,k}^l - \left|\lambda_+^T \times \mathcal{R}_T^l\right| \\ \acute{x}_{m,k}^u = x_{m,k}^u + \left|\lambda_+^T \times \mathcal{R}_T^u\right| \end{cases} \quad \text{where} \quad k \in \{1, 2, \dots, n\} \tag{9.9}$$

Here, λ_+^T is the expansion rate and ranges between $0 < \lambda_+^T \leq 1$. Algorithm 6 summarises the stepwise procedure of SST.

It should be noted that the parameter values should be scaled so that parameters with large values do not disproportionately affect the space-shrinking or expanding process. Scaling is done using Equation 9.10 to avoid negative natural log values, where $[a, b] = [1, 2]$.

$$\mathbf{x}_t \mapsto \frac{\mathbf{x}_t - \mathbf{x}_t^l}{\mathbf{x}_{p_t}^u - \mathbf{x}_t^l} \times (b - a) + a \tag{9.10}$$

To track the amount of shrinkage or expansion for the design space after each interaction, we introduce a quantity $\mathcal{Q}$ (see Equation 9.11), which calculates the average percentage of shrinkage or expansion amount in the T^{th} interaction for the dimensions of the design space.

$$\mathcal{Q} = \frac{1}{n} \times \sum_{k=1}^{n} \left(100 - \frac{\acute{x}_{t,k}^u - \acute{x}_{t,k}^l}{x_{t,k}^u - x_{t,k}^l} \times 100\right) \tag{9.11}$$

Algorithm 6 The pseudo-code of SST

Input: $\mathcal{X}$, N, s and Ω.
Generate an initial design set $\mathcal{N}_{op} \leftarrow GDT(\mathcal{X}, N, s, \Omega)$.
Display designs in $\mathcal{N}_{op}$.
Initialise $T \leftarrow 0$
repeat
  $T \leftarrow T + 1$
  Select t^{th} design ($\mathbf{x}_t$) from $\mathcal{N}_{op}^{T-1}$ (Note: $\mathcal{N}_{op}^{0} = \mathcal{N}_{op}$).
  Input: Ω, N, s, λ_{-}^{T} or λ_{+}^{T}
  for $k = 1$ to n **do**
    if λ_{-}^{T} is define **then**
      $\mathcal{X}^T \leftarrow \begin{cases} \acute{x}_{m,k}^{l} \leftarrow x_{m,k}^{l} + (\lambda_{-}^{T} \times \mathcal{R}_{T}^{l}) \\ \acute{x}_{m,k}^{u} \leftarrow x_{m,k}^{u} - (\lambda_{-}^{T} \times \mathcal{R}_{T}^{u}) \end{cases}$
    else if λ_{+}^{T} is define **then**
      $\mathcal{X}^T \leftarrow \begin{cases} \acute{x}_{t,k}^{l} \leftarrow x_{m,k}^{l} - (\lambda_{-}^{T} \times \mathcal{R}_{T}^{l}) \\ \acute{x}_{m,k}^{u} \leftarrow x_{m,k}^{u} + (\lambda_{-}^{T} \times \mathcal{R}_{T}^{u}) \end{cases}$
    end if
    if $\acute{x}_{m,k}^{u} > x_{m,k}^{u}$ ($\acute{x}_{m,k}^{l} < x_{m,k}^{l}$) **then**
      $\acute{x}_{m,k}^{u} \leftarrow x_{m,k}^{u}$ ($\acute{x}_{m,k}^{l} \leftarrow x_{m,k}^{l}$)
    end if
    $x_{m,k}^{l} \leftarrow \acute{x}_{m,k}^{l}$ and $x_{m,k}^{u} \leftarrow \acute{x}_{m,k}^{u}$
  end for
  $\mathcal{N}_{op}^{T} \leftarrow GDT(\mathcal{X}^{\mathcal{T}}, N, s, \Omega)$
  Display all designs of $\mathcal{N}_{op}^{T}$.
until The user obtains a satisfactory design(s)

In an interaction between the user and GenYacht, the user can select multiple designs. Let the user select two designs, $\mathbf{x}_{t_1}$ and $\mathbf{x}_{t_2}$. Two design spaces, $\mathcal{X}_1^T$ and $\mathcal{X}_2^T$, are then formed. Therefore, two solution sets, $\mathcal{N}_{op1}^T$ and $\mathcal{N}_{op2}^T$, are separately obtained using GDT so that $2 \times N$ designs are shown to the user in the next interaction $(T + 1)$.

In the $(T + 1)^{th}$ interaction, if the user again selects two designs, one from $\mathcal{N}_{op1}^T$ and the other from $\mathcal{N}_{op2}^T$, then again $2 \times N$ designs are created. However, if the user selects a design from $\mathcal{N}_{op1}^T$ ($\mathcal{N}_{op2}^T$), then $\mathcal{N}_{op2}^T$ ($\mathcal{N}_{op1}^T$) is discarded and $\mathcal{X}_1^{T+1}$ ($\mathcal{X}_2^{T+1}$) is formed to create N designs for further interactions. If the user selects two designs from $\mathcal{N}_{op1}^T$ ($\mathcal{N}_{op2}^T$), $2 \times N$ designs are generated and $\mathcal{N}_{op2}^T$ ($\mathcal{N}_{op1}^T$) is discarded.

9.3.3 User-Interface of GenYacht

GenYacht is programmed in a Microsoft Visual Studio platform using the C++ programming language and Parasolid's (a 3D geometric modelling kernel) API functions. A parent hull, shown in Fig. 9.3, is initially stored in the database. During the interactive process, design modification of the parent shape is performed using the parametric design approach proposed by Khan et al. [4]. In this design framework, the overall hull shape is divided into three regions: Entrance-Region (ER), Middle-Region (MR) and Run-Region (RR). Each region is then represented with a set of geometric parameters such as length (L), beam (B), and depth (D). Moreover, the entrance region is further constituted of three more geometric parameters: entrance angle (θ), bow angle (β) and sheer angle (α). The parametric representation of the parent hull can be seen in Fig. 9.3. The description of these parameters with their upper and lower bounds values (in meters) used for the study's experiments are given in Table 9.2.

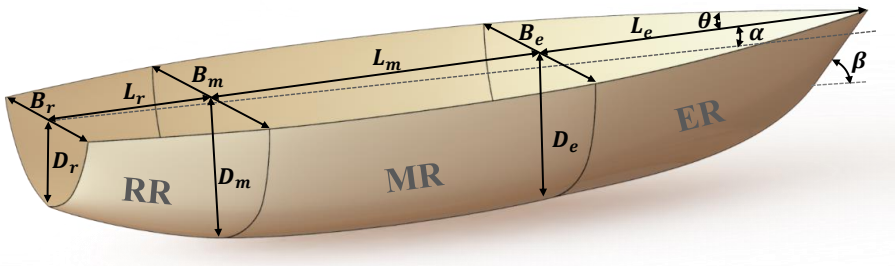


Figure 9.3: Parametric representation of the parent yacht hull created using Khan et al.'s design technique [4]. The parent hull is divided into three regions: Entrance, Middle and Run. An independent set of geometric parameters represents each region.

The main window of GenYacht consists of an OpenGL-based graphical interface for design visualisation (see Fig. 9.4). There are several dialog boxes in GenYacht for interactive designs, calculating hydrostatics and resistance, and setting the initial design space. To start the interactive design, the user first retrieves the parent hull using the 'initial design button' in the main window. The user then inputs the number of designs generated in the interactive design dialog box and selects the geometric parameters. To create a design

Table 9.2: Geometric parameters with their lower and upper bounds for the yacht hull.

Parameters	Definition	[LB, UB] (meters)	Parameter	Definition	[LB, UB] (meters)
L_e	Length of ER	$[5.0, 8.0]$	D_e	Depth of ER	$[2.3, 4.3]$
L_m	Length of MR	$[4.0, 10.0]$	D_m	Depth of MR	$[2.2, 4.2]$
L_r	Length of RR	$[2.0, 6.0]$	D_r	Depth of RR	$[1.7, 3.0]$
B_e	Beam of ER	$[5.0, 7.0]$	θ	Entrance Angle	$[30°, 90°]$
B_m	Beam of MR	$[5.4, 7.4]$	β	Bow Angle	$[30°, 100°]$
B_r	Beam of RR	$[3.4, 5.4]$	α	Sheer Angle	$[0°, 3°]$

where LB: Lower Bound, UB: Upper Bound

space used in the interactive design process, the user can set any values for the upper and lower bounds of geometric parameters using the design space dialog box. GenYacht generates the specified number of yacht hulls, and the physical results of these hulls can be calculated at a user-given draft value and Froude number. Based on the designs' form appearance and physical results, the user next makes design selection(s). Along with the selected design(s), the user inputs the shrink/expand rate value in the interactive design dialog box, which generates designs in the shrunk/expanded space for the next interaction. The user keeps interacting with GenYacht until the desired final designs are obtained.

GenYacht also provides users with the ability to define different geometric constraints at any time during the interactive process(such as overall length (L_{OA}), maximum beam (B_{max}) and maximum depth (D_{max})). The physical constraints can also be implemented in GenYacht to generate a hull with specific performance characteristics. For instance, users can put a constraint on creating designs with specific resistance values. However, care should be taken while defining the constraints and design space, as there can be a case when a hull with a particular performance criterion might not be generated within the given design space. GenYacht notifies the user of the occurrence of such a situation. A user can also export the final design in the **.x_t** file format can be later imported to other digital platforms for further design and performance analysis. Hydrostatics and resistance results of the hull design can also be exported to a **.xlsx** file for future study.

9.4 Results and discussion

In this section, we first compare the performance of five different meta-heuristics while integrating them with GDT. Afterwards, the efficiency of the GDT and SST is demonstrated with various experiments, and the effectiveness of GenYacht is also validated with a user study. Finally, we compared the performance of GenYacht with IGA.

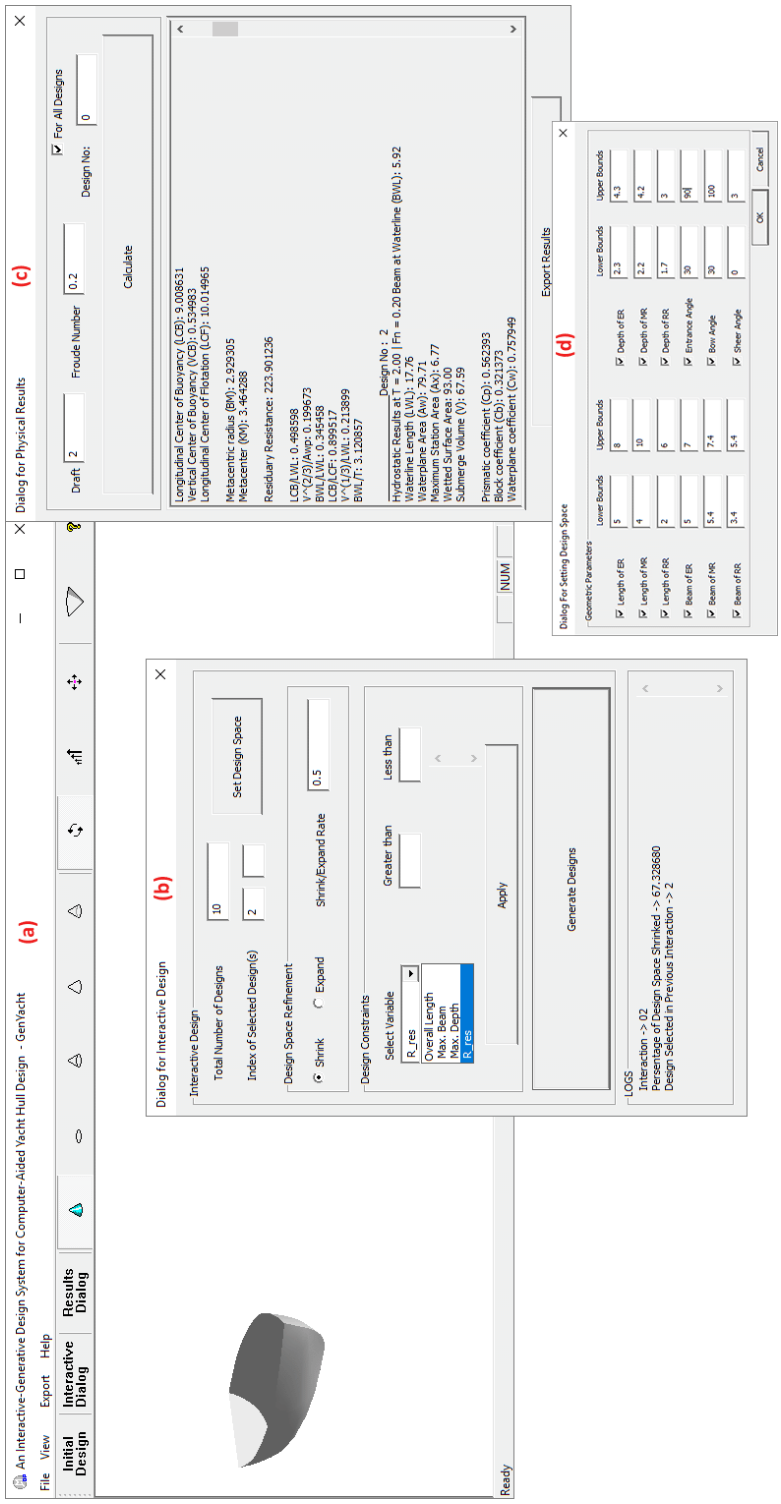


Figure 9.4: The user interface of GenYacht consists of the main window (a), a dialog box for user-GenYacht interaction (b), a dialog box for calculating hydrostatics and resistance (c) and a dialog box for setting the initial design space (d).

9.4.1 Optimiser selection for GDT

Five different optimisation techniques, GA, PSO, ABC, TLBO and JA, were tested in this section. Among them, TLBO and JA are newly proposed yet powerful methods which do not require any parameters to tune their convergence performance. Therefore, this quality of TLBO and JA alleviates an extra burden on the user during the design process. Our aim of testing these optimisation techniques was to select the one having converged to the most negligible value of $\mathbf{F}(\mathcal{N})$ (Equation 9.6) in lesser computational time. As mentioned in [186], an interactive design approach with high computation cost may result in user fatigue. Long waiting times between interaction loops can cause a loss of user interest.

Mutation and crossover rates were set to 0.1 and 0.8 in GA, which controls the exploration and exploitation of the search process. The linear-decreasing-inertia-weight was used, and cognitive (c_1) and social (c_2) learning factors were taken as 2 in PSO. The number of employed and onlooker bees was set to the size of the sub-population (s) in ABC. The performances of these optimisation techniques were tested under the standard algorithmic settings of $N = 10$, $n = 12$, $s = 10$ and $\Omega = 6$. Fig. 9.5 shows a plot between $\mathbf{F}(\mathcal{N})$ and the number of iterations. The computational time taken by the optimisers is given in Table 9.3.

It can be observed from the plot in Fig. 9.5 that JA, GA and TLBO have similar performance, while JA converged to a lower value of $\mathbf{F}(\mathcal{N})$, and it can create completely non-collapsing designs (see Table 9.3). Based on these results, JA was selected to be used in GDT to update the designs in the sub-populations.

9.4.2 Validation of GenYacht system

In this section, the results of GDT and SST will be given, which are embedded in the GenYacht system.

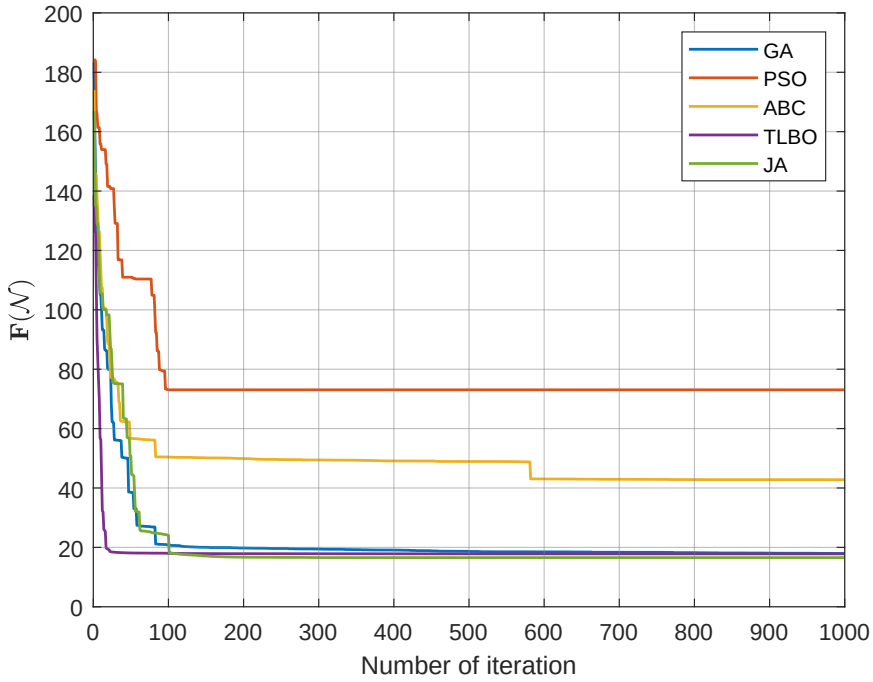


Figure 9.5: Plots for the objective function ($F(\mathcal{N})$) versus the number of iterations performed in GA, PSO, ABC, TLBO and JA.

Table 9.3: Computational times for GA, PSO, ABC, TLBO and JA when used with Algorithm 5

	Computational Time (minutes)	Space-filling (F_1)	Collapsing Designs
JA	**0.84**	**16.56**	**0**
TLBO	3.51	17.82	3
GA	2.14	17.95	**0**
ABC	1.40	18.79	3
PSO	2.94	19.06	6

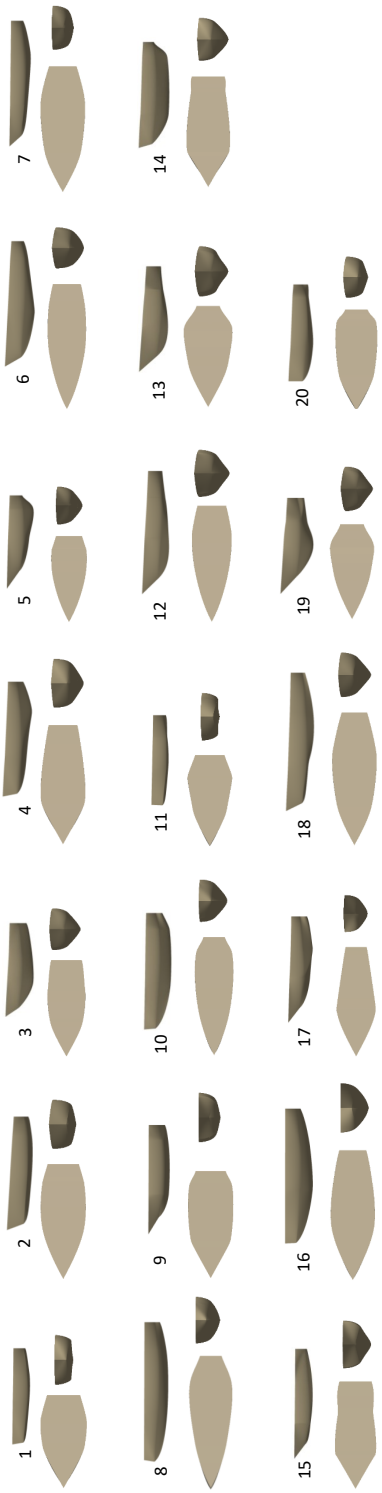


Figure 9.6: Design alternatives generated using GDT for the hull model in Fig. 9.3 (For better visualisation of designs in this figure, the reader is referred to the digital version of this article).

Results of GDT

Fig. 9.6 shows 20 space-filling design alternatives for the parent hull, which were generated using GDT. These alternatives are searched within a 12-dimensional design space bounded with parametric limits shown in Table 9.2 and with the parameter settings of $s = 10$ and $\Omega = 6$. From Fig. 9.6, one can easily observe that the designs are distinct, which can help users of GenYacht to start the interactive process with a design that meets their requirements. As mentioned before, the design modification is performed using Khan et al.'s parametric design framework [4], which locally modifies the geometric parameters. To ensure the generation of plausible shapes (i.e., realistic hull shapes) in interactions, the following *hard design constraints* have been implemented: (1) $-\frac{B_e}{3} \leq (B_e - B_m)$, (2) $-\frac{D_e}{3} \leq (D_e - D_m)$, (3) $B_r \leq (B_m, B_e)$ and (4) $D_r \leq (D_m, D_e)$. The first two constraints limit the parameters B_m and D_m, and the last two constraints B_r and D_r. According to our experience, if $B_m >> B_e$ or $D_m >> D_e$, and if $B_r >> (B_m, B_e)$ or $D_r >> (D_m, D_e)$, implausible designs can occur as shown in Fig. 9.7 (a) and (b), respectively.

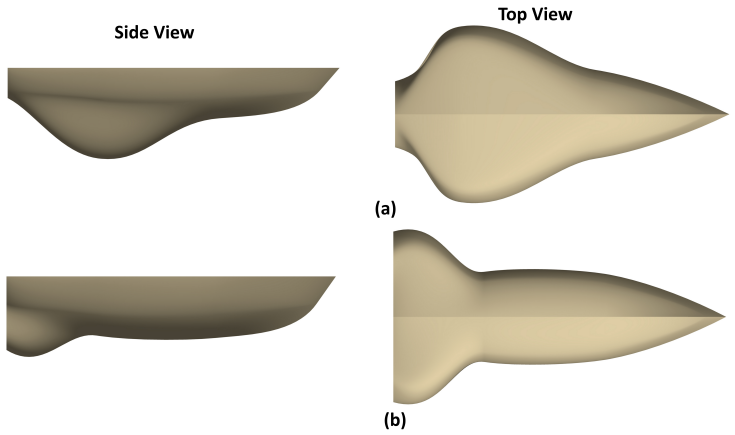


Figure 9.7: Example of implausible/non-realistic designs without hard design constraints.

During the hull form design, there are a variety of numerical performance analyses, consisting of both hydrostatics and hydrodynamics, that naval engineers have to perform to determine whether the hull form can fulfil the design requirements before the selection

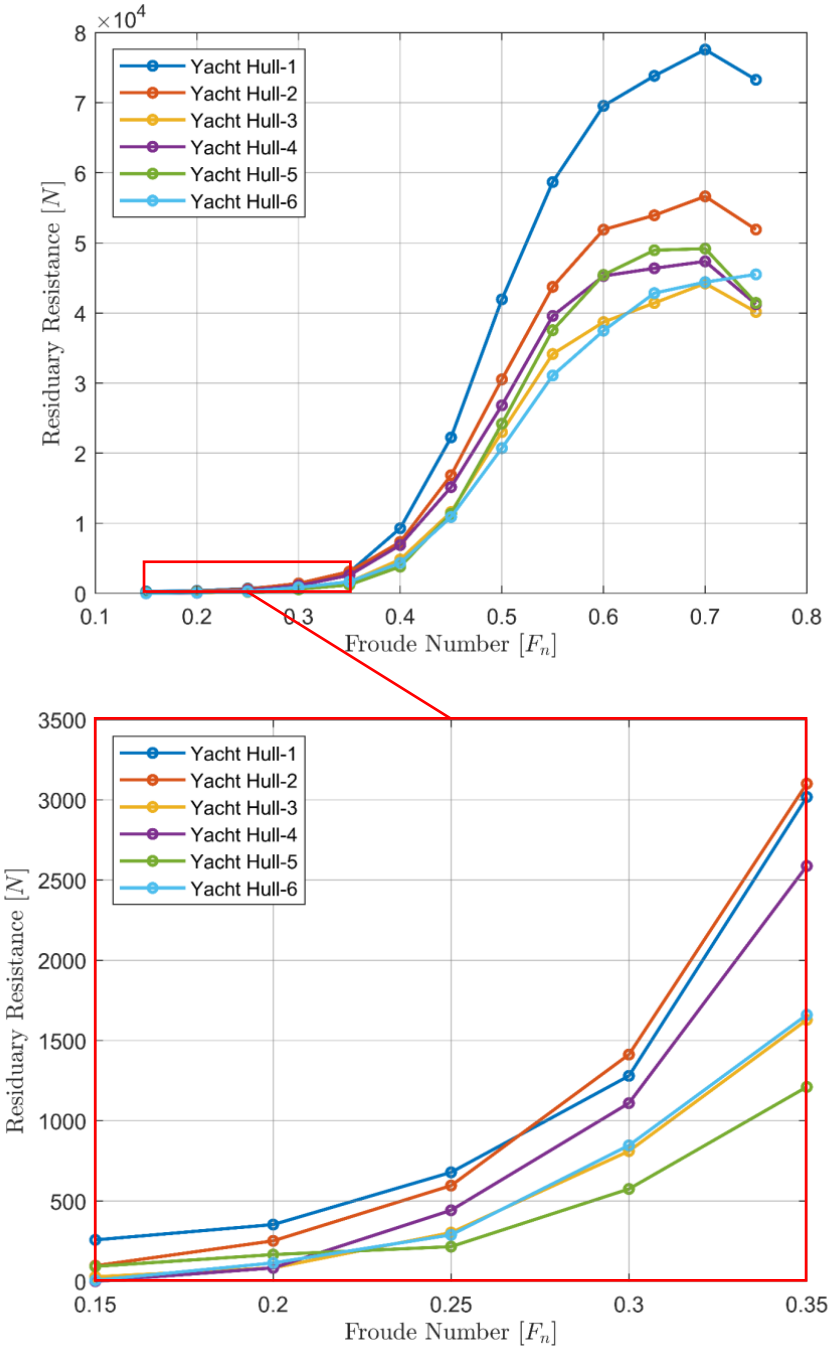


Figure 9.8: Plot showing the residuary resistance (R_{res}) versus Froude number (F_n) for the first six designs in Fig. 9.6.

Table 9.4: Hydrostatics and resistance results of the first ten design alternatives shown in Fig. 9.6.

Hull	1	2	3	4	5	6	7	8	9	10
D	2.54	2.95	4.00	3.99	3.61	3.68	3.57	3.18	4.02	2.50
T	1.00	1.00	1.50	1.50	1.50	1.50	1.50	1.50	1.50	1.50
F_n	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20
BWL	6.83	6.35	4.11	5.25	4.37	4.87	5.79	6.70	4.28	6.47
LWL	14.78	16.96	13.04	16.96	10.08	17.33	21.04	14.59	17.58	14.79
A_{wp}	70.31	75.92	36.96	60.14	31.95	58.90	78.29	77.61	48.61	62.72
A_x	5.18	4.15	3.43	3.93	3.81	4.22	5.42	7.15	3.80	4.22
A_{ws}	77.38	80.31	46.64	67.67	37.28	69.03	92.56	88.64	63.60	69.98
V	42.31	38.90	26.46	33.60	20.26	37.06	63.10	72.09	37.75	36.75
IT	200.35	194.14	36.03	92.48	37.10	84.76	147.99	233.29	52.26	145.05
LCB	8.41	10.09	6.36	9.45	6.25	9.05	11.70	7.97	10.10	8.55
KB	0.34	0.32	0.48	0.40	0.44	0.44	0.50	0.54	0.51	0.32
LCF	8.30	9.44	6.66	8.52	5.82	10.03	11.72	7.87	9.98	8.44
BM	4.74	4.99	1.36	2.75	1.82	2.29	2.35	3.24	1.38	3.95
KM	5.07	5.30	1.84	3.15	2.67	2.72	2.85	3.77	1.90	4.27
C_p	0.55	0.55	0.59	0.50	0.53	0.51	0.55	0.69	0.56	0.59
C_b	0.42	0.36	0.33	0.25	0.31	0.29	0.35	0.49	0.33	0.40
C_{wp}	0.70	0.70	0.69	0.68	0.73	0.70	0.64	0.79	0.65	0.66
C_m	0.76	0.65	0.56	0.50	0.58	0.58	0.62	0.71	0.59	0.69
R_{res}	353.10	251.74	83.01	145.27	166.24	114.05	225.99	417.82	121.06	282.74
R_F	575.80	663.84	315.53	559.36	207.67	580.12	903.01	653.12	310.98	521.01
R_T	928.90	915.58	398.54	704.63	373.91	694.17	1129	1070.94	432.04	803.75

where D: Depth (m), T: Draft (m), BWL: Width at waterline (m), LWL: Length at waterline (m), A_w: Waterplane area (m^2), A_x: Maximum sectional area (m^2), A_{ws}: Wetted surface area (m^2), V: Volume (m^3), IT: Transverse moment of inertia (m^4), LCB: Longitudinal center of buoyancy (m), KB: Vertical center of buoyancy (m), LCF: Longitudinal center of flotation (m), BM: Metacentric radius (m), KM: Metacenter height (m), C_p: Prismatic coefficient, C_b: Block coefficient, C_{wp}: Waterplane coefficient, C_m: Midship coefficient, R_{res}: Residuary Resistance (N), R_F: Frictional Resistance (N) and R_T: Total Resistance (N).

of the final design. Therefore, using GenYacht, users can also evaluate the hydrostatics properties, form coefficients, residuary (R_{res}) and frictional (R_F) resistance of each hull design at the given values of the draft (T) and Froude numbers (F_n). Table 9.4 shows the hydrostatics and resistance results for the first ten designs in Fig. 9.6. It should be noted that the hydrodynamics of a hull includes wave resistance, sea-keeping, manoeuvrability, and so forth, which mostly require Computational Fluid Dynamic (CFD) analyses to be performed. However, running these computationally expensive analyses makes the user wait long before performing the next interaction. This can also result in directing the exploration process towards the non-preferred regions. Therefore, we have utilised empirical equations proposed by Keuning and Katgert [220] to calculate the R_{res} of the hull alternatives. Fig. 9.8 shows the plots of R_{res} (expressed in Newton) versus F_n of the first six designs in Fig. 9.6. The differences in the appearances and performances of the designs in the plots of Fig. 9.8 demonstrate that the designs generated by the proposed system in Fig. 9.6 are diverse in terms of both appearance and performance. Fig. 9.8 also validates the implementation of Keuning and Katgert's [220] technique to calculate residuary resistance at different Froude numbers. Frictional resistance is also calculated according to the ITTC formula [221]. Reynolds number (R_n) and frictional resistance coefficient (C_F) are calculated for a yacht navigating in seawater at $15°C$ with density and kinematic viscosity of 1.189×10^{-6} (m^2/s) and 1026.021 (kg/m^3), respectively. The total resistance (R_T) is the sum of R_{res} and R_F.

The core objective of this work is to propose an interactive design system that allows users to generate yacht hull designs at the preliminary stage while taking its form appearance and physical properties into account. After selecting the desired hull form(s), the user can export it and perform detailed hydrodynamic and structural analyses using off-the-shelf computational tools.

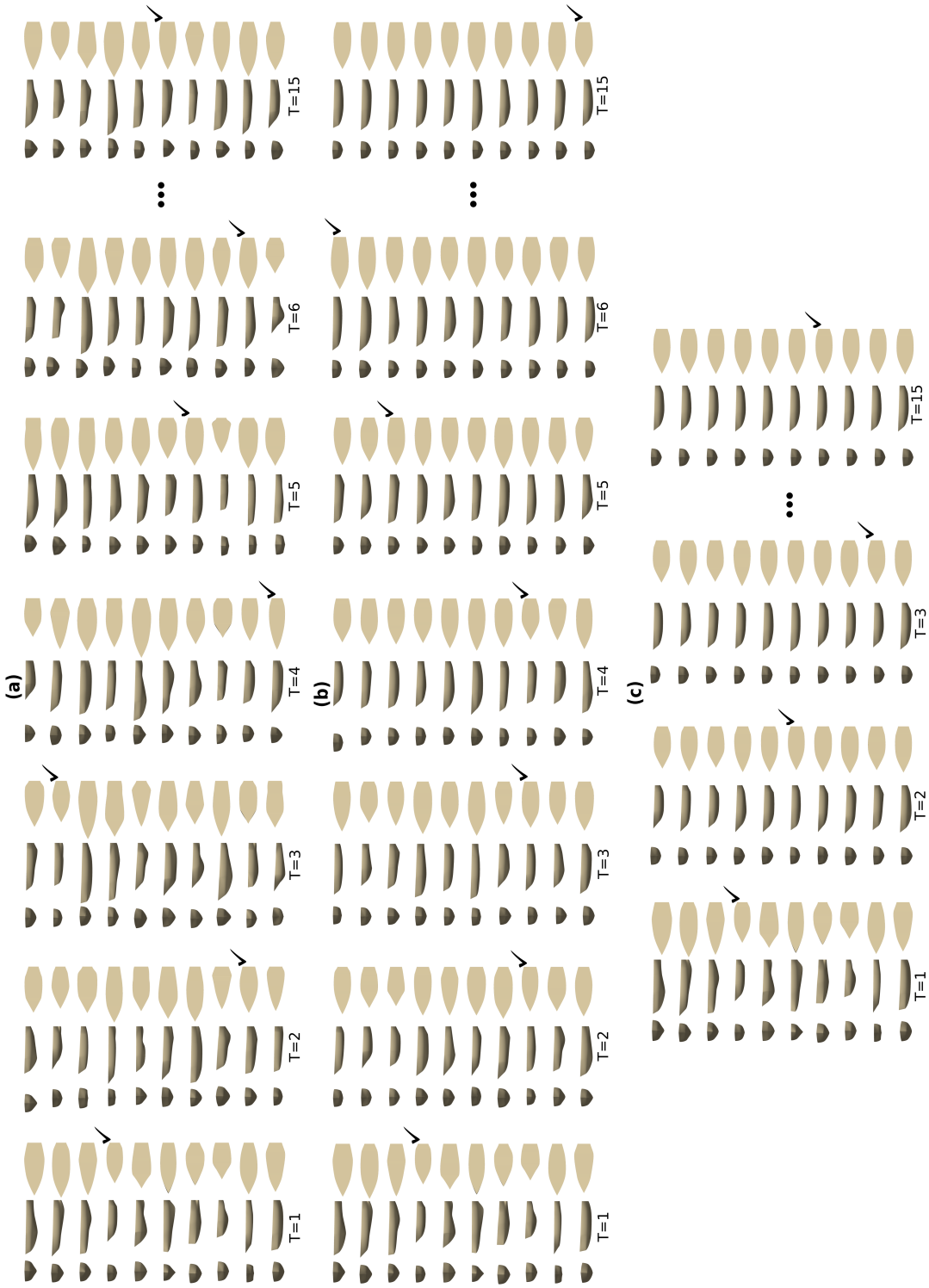


Figure 9.9: Yacht hull alternatives created during fifteen design interactions using a shrink rate (λ_-) of (a) 0.1, (b) 0.5 and (c) 1.0 (For better visualisation of designs in this figure, the reader is referred to the digital version of this article).

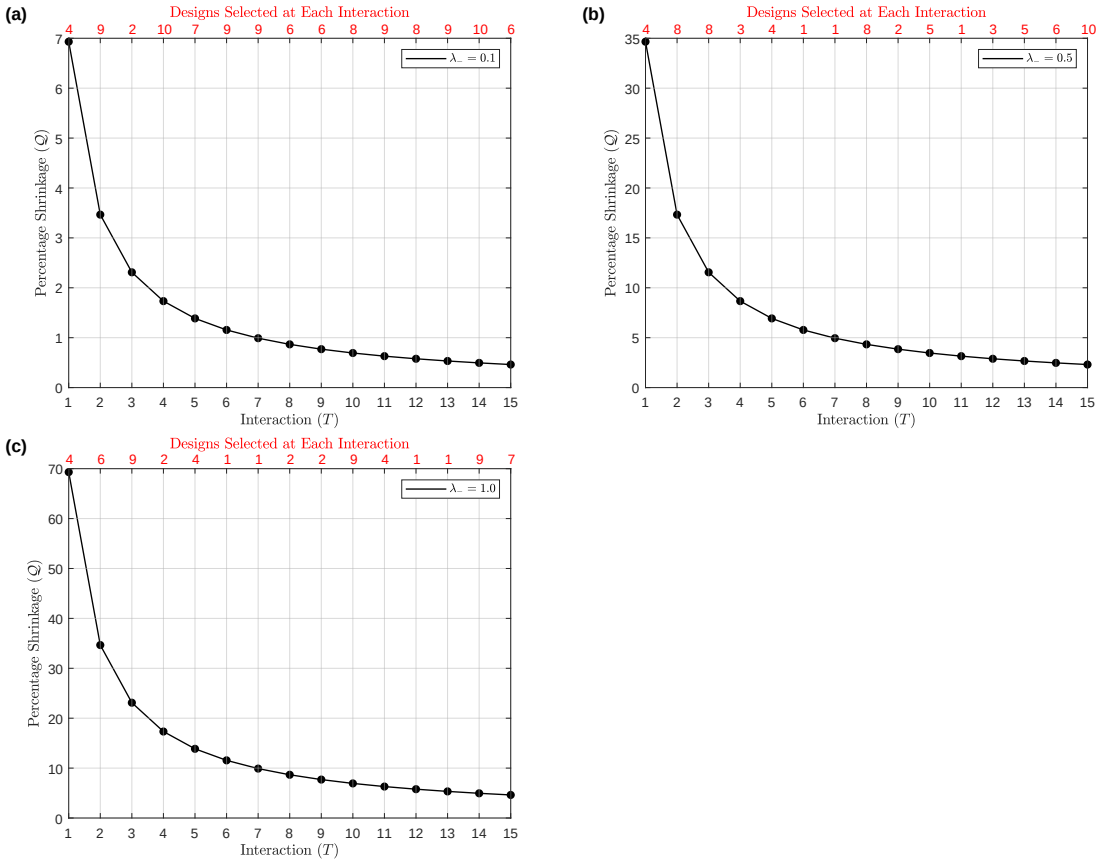


Figure 9.10: Plots showing the percentage shrinkage ($\mathcal{Q}$) of the design space during fifteen design interactions (T) when (a) $\lambda_- = 0.1$, (b) $\lambda_- = 0.5$ and $\lambda_- = 1.0$.

Results of SST

The results of SST were validated with different experiments using different values of the shrink rate (λ_-). Ten design alternatives were first generated, and interaction then proceeded with an objective to select a design having a trade-off between appearance and performance. Fig. 9.9 (a), (b) and (c) show the designs created in the fifteen design interactions with λ_- settings of 0.1, 0.5 and 1.0, respectively. Fig. 9.10 (a), (b) and (c) shows plots for the average percentage of space-shrinkage $\mathcal{Q}$ versus the design interactions (T) in Fig. 9.9 (a), (b) and (c), respectively. In Fig. 9.10, the top axis (in red colour) shows the design selected in each interaction for design in 9.9. The area under the curve represents the percentage of the space shrunk in fifteen interactions. It is noteworthy that at higher values of λ_-, the value of $\mathcal{Q}$ is high in the interactions. For instance, when λ_- was set to 0.1, 0.5 and 1.0, the original design space shrunk by 6.93%, 34.66% and 69.31% in the first interaction ($T = 1$), respectively. Afterwards, in the second interaction ($T = 2$), 3.47%, 17.32%, and 34.66% per cent of the design space created in the first interaction was shrunk.

As mentioned before, at each interaction user selects a design and, depending on the shrink/expend rate value, the design space is shrunk/expended, and new N designs are generated for the next interaction. At higher values of λ_-, the amount of design space shrinks is higher (see Fig. 9.10 (c)), which might create a narrower design space. The designs generated from this space for the next interaction can be similar (i.e. designs with less diversity, see designs in Fig. 9.9 (c)). When λ_- is set to higher values during interactions, the designs converge faster (i.e., get similar) towards the selected design. For instance, the interaction results shown in Fig. 9.9 (c) were obtained using $\lambda_- = 1.0$. In this setting, designs started to converge after the third interaction.

On the contrary, designs generated from a design space created with a smaller value of λ_- will be more diverse. As shown in Fig. 9.10 (a), when λ_- was set equal to a minimal value (i.e.$\lambda_- = 0.1$), the shrinkage of the design space in each interaction is small, and

designs generated are diverse. Thus, it may require a higher number of iterations for a user to converge to the final design. However, for small values of λ_{-}, the user can explore more variety of designs. For instance, the design generated in Fig. 9.9 (a) are created when $\lambda_{-} = 0.1$; therefore, even at the 15^{th} interaction, designs are still diverse, thereby showing a slow convergence. Moreover, as shown in Fig. 9.9 (b), at $\lambda_{-} = 0.5$, the user can achieve a better trade-off between design diversity and convergence because the amount of space shrinks at each interaction is moderate (see Fig. 9.10 (b), which shows the amount of space shrinks at each interaction when $\lambda_{-} = 0.5$). Therefore, we recommend the users start the interaction at $\lambda_{-} = 0.5$. Fig. 9.11 shows a plot between $\mathcal{Q}$ and λ_{-}, which confirms a linear relationship between these two parameters.

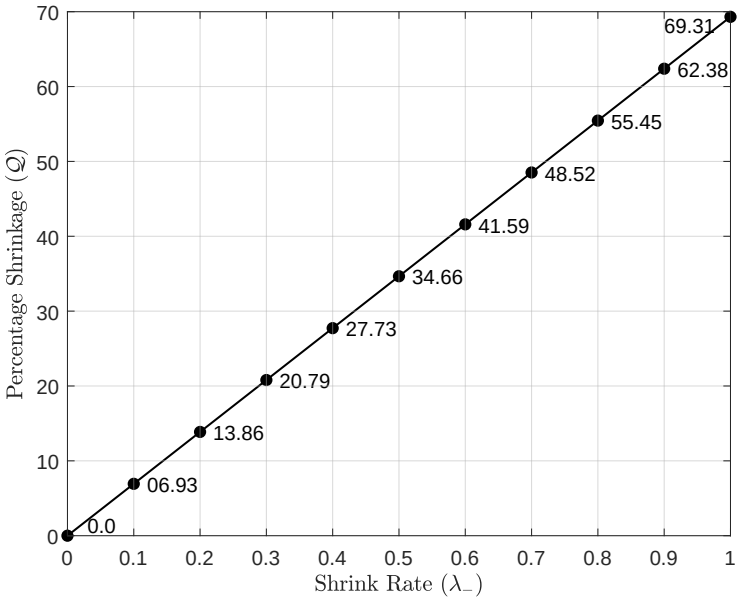


Figure 9.11: Plot showing the percentage shrinkage ($\mathcal{Q}$) of the design space versus shrink rate (λ_{-}).

The ability of the proposed system to search for a target design was also tested. First, a target design was randomly selected from the design space in Table 9.2 and its parameter values and hydrostatic properties were stored. Afterwards, the interactive process was

started to replicate the target design. At the first interaction, 20 designs were generated, and from these designs, a design having parameter and hydrostatics values close to the target one was selected. Based on the selected one, 20 new designs were generated, and the process was repeated for four interactions. The target and final design obtained after the fourth interaction is shown in Fig. 9.12. It can be observed that visually both designs are very similar. Moreover, their parameter values and hydrostatic properties, shown in Table 9.5, are also close to each other. This validates the ability of the GDT and SST techniques to converge to the desired hull design.

Here, the interactive process should be started with an appropriate number of designs to visualise all the uniformly-distributed designs that sufficiently cover the design space. However, this number can be high, particularly for the high-dimensional design spaces or the design spaces whose dimensional bounds are large. The claim that the proposed method removes user fatigue may not hold true in such cases. However, as proven via experiments, using GenYacht, the user can still explore a design space well compared to IGA.

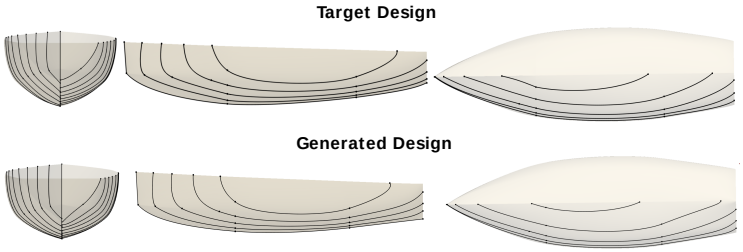


Figure 9.12: The design space is explored interactively to replicate a target design. The image at the top is the target design, and the image at the bottom is the design generated after four interactions using GenYacht. The similarity between the two designs indicates that the user could approximate the target design well.

9.4.3 Computational time

As mentioned before, one of the crucial criteria in interactive design techniques is that it should be computationally less expensive. Long waiting times cause user fatigue, thereby

Table 9.5: Parametric values and hydrostatic properties of the design shown in Fig. 9.12. Hydrostatic properties were calculated at the draft of 2.0 meters.

Parameters (units)	Target Design	Generated Design
LOA (m)	20.72	20.07
B_{max} (m)	6.62	6.84
D_{max} (m)	3.49	3.71
A_{wp} (m^2)	92.26	92.69
A_x (m^2)	7.89	7.38
A_{ws} (m^2)	116.23	113.12
V (m^3)		96.06
IT (m^4)	205.70	193.46
LCB (m)	11.20	11.17
KB (m)	0.670	0.645
LCF (m)	11.40	11.56
BM (m)	1.98	2.05
KM (m)	2.65	2.70
C_p	0.636	0.608
C_b	0.414	0.382
C_{wp}	0.737	0.754
C_m	0.651	0.629

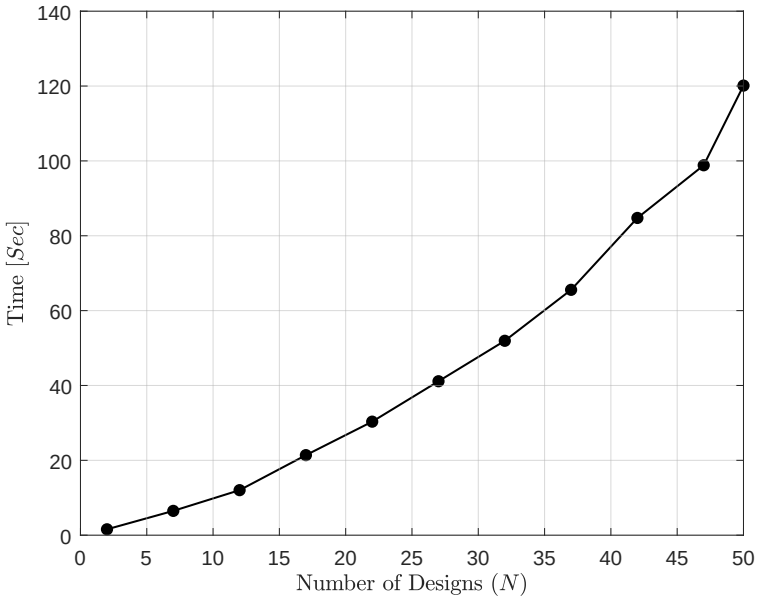


Figure 9.13: Plot showing GenYacht's computational cost (in seconds) versus a number of designs (N).

hindering the user from effectively exploring the design space for a good design. The experiments in this study were conducted using a PC with an i7-7700 Intel Core, 3.6-GHz processor, and 8 GB physical memory. Fig. 9.13 shows a plot of the computational time (in seconds) of GenYacht versus the number of designs (N) generated. The computational cost is the sum of the computational time taken by GDT to explore N space-filling designs, parametric modification of N designs using Khan et al.'s [4] approach, computation of hulls' physical properties and space-refinement in one interaction. From Fig. 9.13, it can be seen that GenYacht took approximately two minutes to create 50 designs in an interaction. These results confirm that the computational complexity of the proposed system is significantly low.

Figure 9.14: The designs generated by the subjects in the user study (For better visualisation of designs in this figure, the reader is referred to the digital version of this article).

9.4.4 User study

A user study was conducted to validate the efficiency and feasibility of GenYacht. We selected ten PhD candidates as the subjects in the user study from the Department of Naval Architecture, Ocean and Marine Engineering at the University of Strathclyde, who had an average 3.60±2.67 (average ± standard deviation) years of industrial and research experience in the ship and parametric design. A brief introduction of the interactive designs approach was first presented to the subjects, and a small training session on the proposed system was then given along with a description of SST and its behaviour with the shrink rate. Plots in Fig. 9.10 and 9.11 were described to them, so they could better understand tuning this parameter. Subjects were also familiarised with the geometric parameters of the primary hull form, and they were asked to set some design specifications before starting the interactive process. To avoid user fatigue, we asked subjects first choose some designs based on their form appearance and then compare these designs based on their physical performance before making the final selection or vice versa. The results of the interactive process for each subject are shown in Table 9.6, and the designs generated by the subjects are shown in Fig. 9.14. The average time the subjects completed the interactive process was 5.12±1.07 minutes.

After the interactive process was completed, we asked the following questions to the subjects for further evaluation of the system. Their responses were acquired on a 5-point Likert scale (1: Strongly Disagree, 2: Disagree, 3: Neutral, 4: Agree, 5: Strongly Agree):

Q1: GenYacht is easy to use in an interactive generation of hull forms.

Q2: GenYacht yacht provides a more sophisticated approach for preliminary hull design compared to the traditional parametric design exploration techniques.

Q3: Using GenYacht, I was able to generate a satisfactory design within my design requirements.

The average Likert scores given by the subjects for the first, second and third questions

Table 9.6: Results of the user study.

Interaction	Subjects	1	2	3	4	5	6	7	8	9	10
	Initial Designs (N)	15	25	17	10	10	25	21	17	15	20
	Design Selected	8	24, 7	3	1	9	20	9	15	8	9
$T = 1$	λ_{-}^{1}	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.3	0.7	0.5
	N^1	10	10	17	13	10	5	21	17	15	20
	Design Selected	7	19	1	4	8	1	15	2	10	6
$T = 2$	λ_{-}^{2}	0.5	0.5	0.5	0.8	0.5	0.5	0.8	0.5	0.8	0.3
	N^2	7	10	10	6	10	10	21	17	15	20
	Design Selected	3	1	4	1, 2	5	1	8	15	**9**	12
$T = 3$	λ_{-}^{3}	0.5	0.5	0.5	0.8	0.7	0.9	1.0	0.6	-	0.3
	N^3	5	5	5	3	10	5	21	20	-	15
	Design Selected	3	2	1	3	6	**1**	**15**	9	-	**13**
$T = 4$	λ_{-}^{4}	0.5	0.5	1.0	1.0	0.9	-	-	0.6	-	-
	N^4	5	5	5	6	20	-	-	20	-	-
	Design Selected	**2**	**2**	4	**4**	7	-	-	2	-	-
$T = 5$	λ_{-}^{5}	-	-	1.0	-	0.9	-	-	0.8	-	-
	N^5	-	-	3	-	5	-	-	20	-	-
	Design Selected	-	-	**3**	-	**1**	-	-	2	-	-
$T = 6$	λ_{-}^{6}	-	-	-	-	-	-	-	1.0	-	-
	N^6	-	-	-	-	-	-	-	20	-	-
$T = 7$	Design Selected	-	-	-	-	-	-	-	**3**	-	-

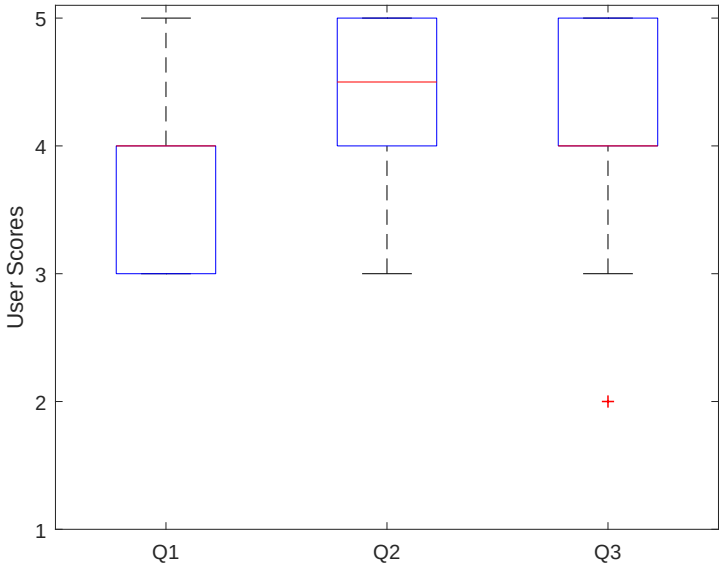


Figure 9.15: Box and Whisker plot the scores given to questions Q1, Q2 and Q3 given by the subjects during the user study. The subjects were asked these questions at the end of the user study for further evaluations of GenYacht.

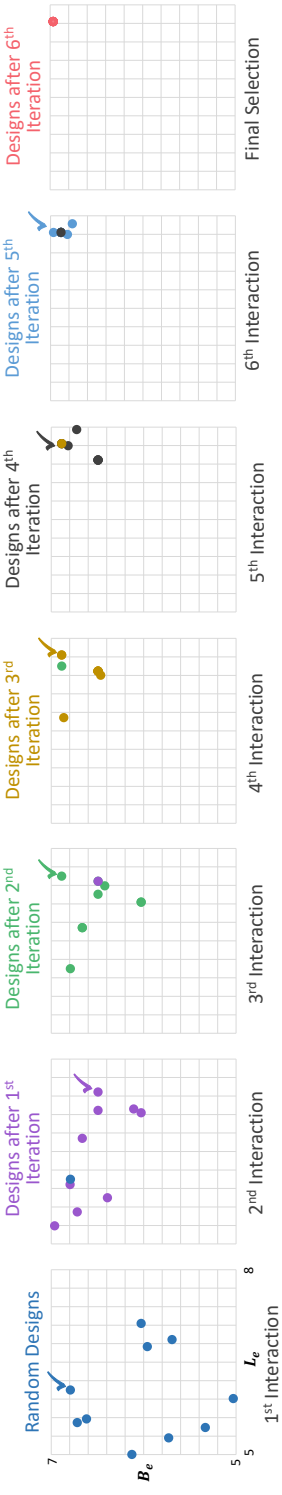


Figure 9.16: Interactive results of the interactive genetic algorithm (IGA). The design space was created using two geometric parameters, L_e and B_e. Designs were first created for the first interaction. The genetic algorithm (GA) then performed an iteration to generate a new population for the subsequent interaction while converging towards the selected design. The interactive process was repeated until all the designs converged to the preferred one (see the last image).

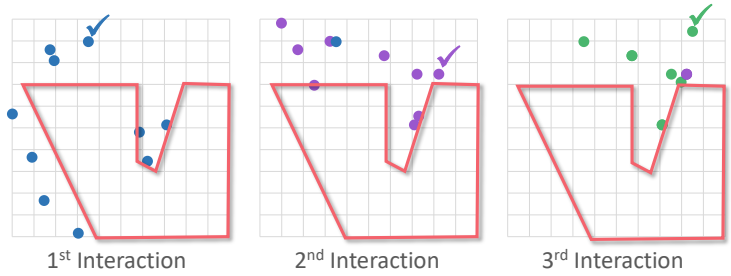


Figure 9.17: The area bounded in red (of the design space in Fig. 9.16) was unexplored when IGA was used.

were 3.90±0.7379, 4.40±0.6992 and 4.00±0.9428, respectively. The variations of the user scores were also analysed using the Box and Whisker plot, shown in Fig. 9.15. These results indicate that users could generate good designs using GenYacht. Some subjects also suggested creating parent shapes using GenYacht, which they would like to further optimise for the specific performance criteria. Subjects also like that GenYacht allows users to compare a wide variety of designs, which is essential in ship design because mostly the optimal configuration is the one that best satisfies the customers' design requirements.

9.4.5 Comparison with IGA

We have also compared the performance of GenYacht with an IGA-based technique. As mentioned in Section 9.2.1, there are many variations of IGA in literature. In this work, we implemented IGA similar to [186] and utilised first a two-dimensional design space formed using the geometric parameters, L_e and B_e, of the parent hull for better visualisation of GenYacht's and IGA's performances (see Figures 9.16 and 9.18). However, we selected a preferred design instead of rating the designs as this selection scenario is similar to GenYacht. An initial population of random designs was first generated, and a design was selected from the initial population. GA then performed an iteration/generation to create a new population to minimise the normalised Euclidian distance between the chosen design and designs in the population pool. In each generation, designs were evaluated to make a selection. Here, GA was used with crossover and mutation rates of 0.8 and 0.1,

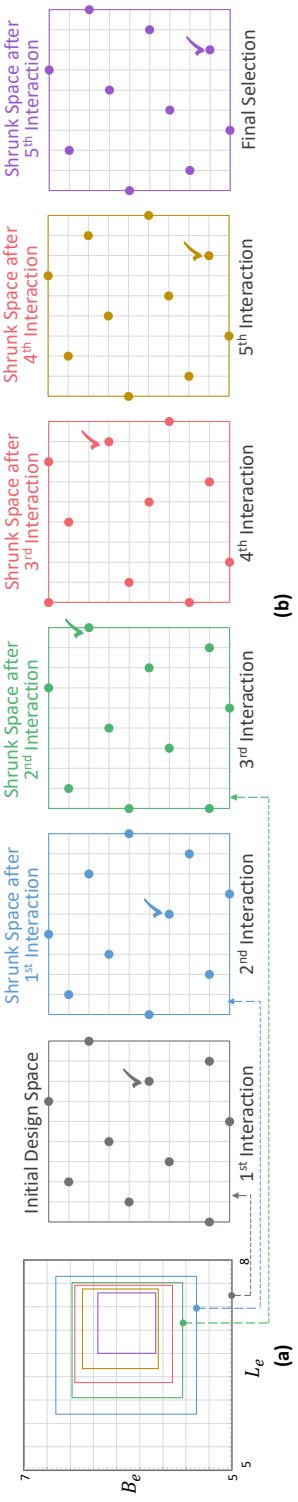


Figure 9.18: Interactive results when using GenYacht in the design space shown in Fig. 9.16. Initial and shrunk design spaces in each interaction (a). Ten designs were generated using GDT in the interactions, and the design space was shrunk using the space shrinking technique (SST) based on the user selection at shrink rates of $\lambda_- = 0.5$ for the first three and $\lambda_- = 1.0$ for the last two interactions (b).

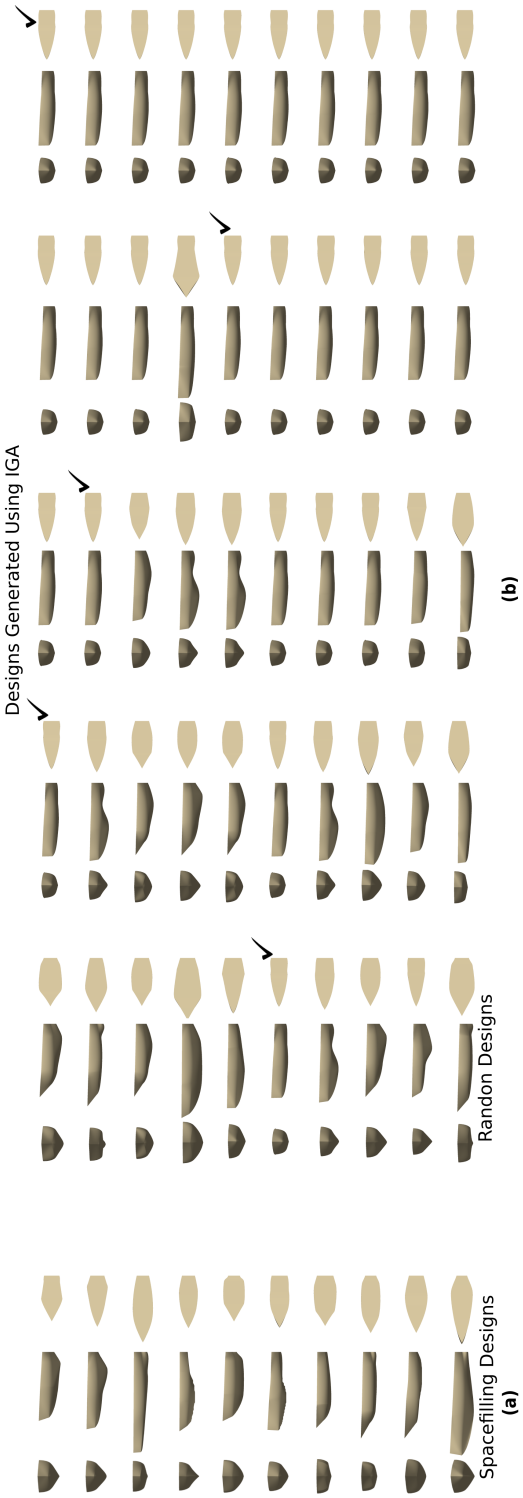


Figure 9.19: Hull forms generated using the GDT (a). Hull forms were created during the interactive process using IGA (b). Designs converged (i.e., got similar) at the fourth interaction without significant diversifications in the hull models. Moreover, the left-most image of (b) shows the randomly generated designs, which are similar to each other compared to designs generated using GDT.

respectively. It can be seen from Fig. 9.16 that after each generation of GA, the newly generated designs moved towards the selected design. Here, it is noteworthy that IGA focuses on the convergence towards the selected hull form instead of maintaining diversity.

Furthermore, the main drawback of IGA is that it depends mainly on the initial population. Therefore, starting the interactive process with randomly sampled designs may limit exploration. More than 50% of the design space remained unexplored when IGA was used in Fig. 9.16. Fig. 9.17 highlights these unexplored regions.

Fig. 9.18 (b) and (a) show the interactive results of GenYacht for a two-dimensional design space created with L_e and B_e, and the shrunk design space at each interaction. It can be observed that, compared to IGA, GenYacht let the users start the design process with well-sampled diverse designs, and explores the design space effectively at each interaction. Moreover, SST provides a sophisticated way to focus the computational effort on exploring potential regions.

Fig. 9.19 (a) shows the yacht hull designs generated using GDT, and the left-most image of Fig. 9.19 (b) shows the randomly sampled designs, which were then used to perform the interaction process using IGA. The space-filling values ($F_1(\mathcal{N})$) of GDT and random designs is 17.2708 and 26.1553, respectively. The high value indicates that the random design does not spread in the design space evenly. A glance at the appearance of these designs can reveal that there exists a clustering pattern in randomly generated hull forms (first image of Fig. 9.19 (b)). For instance, from the top, the first three, the next four and the last three designs are similar. However, designs created via GDT (Fig. 9.19 (a)) are unique to a large extent. Fig. 9.19 (b) also shows the interactive results of IGA. It can be observed that the designs generated in each interaction are very similar as there is no control for the user maintaining the design diversity.

9.5 Conclusions and future works

This work proposes a novel interactive and generative design-based CAD system for the preliminary design of yacht hull forms. The proposed system introduces a new design approach in naval architecture, which enables naval architects, engineers and novice users to integrate their design preferences about the hull form into the design of space exploration. Users can generate designs that best fit their design requirements, not only in terms of physical performance but also considering the design's overall appearance. In GenYacht, a generative design approach first generates a user-defined number of space-filling hull forms satisfying the given design constraints. Among these designs, the user selects a suitable one, which is then used to create a new design space using a space-shrinking technique. The new space is then fed to the generative design technique to generate a set of space-filling designs for the next interaction. This generative and interactive process continues until the user reaches the desired shape. Experimental and user study results reveal that the proposed system has the potential to create user-centred yacht hull forms, which better reflect designers' design considerations. The new system also benefits the users in naval architecture and marine engineering compared to the parametric-based exploration techniques.

In future work, we plan to use non-dimensional parameters to define the hull design space and to test GenYacht with this design space. We would also like to integrate more physical performance criteria, such as sea-keeping and stability. Furthermore, we would like to develop empirical equations for these criteria using deep learning. Our efforts will also continue to develop a web-based user interface to give better usability to potential users. It will also be worth working on developing a similar interactive system for other types of marine vessels, such as chined hulls (or planing crafts) and multihulls.

Chapter 10

Conclusion

As it has been extensively discussed throughout the thesis that to date, in many industrial sectors, including maritime, designers and engineers use extensively off-the-shelf parametric modellers and simulation tools. These tools are characterised by conservatism, for they are built to generate shapes lying in the neighbourhood of a successful baseline/parent shape. Next, these modellers are coupled with optimisers for improving the baseline shape against performance criteria (e.g., ship wave resistance, sea-keeping, structural strength, etc.), which involve time-consuming simulations, e.g., computational fluid dynamics (CFD). At the end of the process, the new design is likely a local optimum whose shape is a minor variation of the existing one. Conclusively, the coexistence of conservative parametric modellers with high-cost simulations and a large number of design parameters needed for shape optimisation of complex shapes leads to *non-efficient simulation-driven design* pipeline that suffers from the curse of high-dimensionality and a limited capability to explore design spaces efficiently for delivering variant, innovative, user-centred and truly optimal designs.

In this thesis, the objectives of efficient design space exploration and reduction of computational cost are achieved by lowering the design space's dimensionality i) by eliminating the parameters which are less sensitive/significant towards the physical QoI using PSA and ii) by extracting the latent feature with feature extraction/embedding approaches to form a

subspace of reduced dimensionality. To achieve this objective, first, a novel intra-sensitivity concept is proposed to study the local behaviour of parametric sensitivities and eliminate instabilities - a parameter can be sensitive in some local regions of the design space but become insensitive in others. Such behaviour makes PSA vulnerable to fluctuations even with slight perturbation in the parametric ranges of the design space. Therefore, the outcome of intra-sensitivity allows designers to construct viable design spaces for the reliable execution of PSA.

Afterwards, to release the computational burden inherited from implementing PSA or intra-sensitivity, a new geometric-moment dependent PSA is proposed that harnesses the geometric variation of designs in a design space using geometric moments as a geometrical QoI to measure parametric sensitivities. These results can be a prior estimation of parametric sensitivities and used to construct a design space of lower dimension with only a subset of highly/strongly sensitive parameters. This approach can significantly reduce the computational time because, typically, sensitivities are learnt directly with physical QoI, which can add a heavy computational burden on the entire design process as one has to perform computationally intensive physical simulations for both PSA and shape optimisation.

The commonly used feature extraction approaches for dimension reduction usually generate subspaces that often fail to accommodate physical information and preserve the intrinsic geometric structure of the shape, thus resulting in many invalid geometries with slow convergence, mostly toward local optima. Therefore, a shape-supervised subspace is developed, which extracts a high-level geometry description as a shape signature vector (SSV) and uses it as a substitute for physics. SSV enables the subspace to preserve the intrinsic geometric structure and embeds latent features related to the designs' physics.

A feature-to-feature learning strategy is also proposed to create a functionally-active subspace for expediting the construction of surrogate models at an off-line stage. To achieve this, we first extract geometrically-active features to capture the features materialising maximum geometric features; afterwards, we use the active-subspace method to remove the

physically-active features from the geometrically-active subspace. The resulting subspace is geometry and physics augmented to construct efficient surrogate models using Gaussian process regression with only a handful of samples. These surrogate models are later used to bypass physical simulations during shape optimisation.

For the versatile parameterisation of ship hulls, we developed ShipGAN using deep convolutional generative adversarial networks, so the resulting parametric modeller is generic with the ability to perform feasible and plausible design modifications for a large variety of hulls. To train the ShipGAN model, we first select 13 different classes of ships, including containers, oil tankers, bulk carriers, naval and crew supply vessels, etc. This approach breaks the current conservatism in the parametric computer-aided ship design paradigm, where parametric modellers can only handle a particular ship hull type. We have shown through extensive experimentation that ShipGAN can create designs with augmented features resulting in versatile design spaces that give geometrically valid and practically feasible shapes.

Finally, we proposed a generative and interactive design tool, GenYacht, which empowers experienced and novice designers to generate all desired optimisation and performance alternatives for boilerplate 'parent' hulls. GenYacht's generative approach samples the design space using a space-filling and non-collapsing optimisation strategy. Minimisation of these criteria ensures that various features are uniformly distributed within the design space. Interactive modelling allows designers to perceive the 3D representation of the hull during the generative process, capturing the targeted regions of change. Generated designs are presented to the customer, and attribute options are selected and combined based on the requirements. Afterwards, the design space is further refined using a novel space-shrinking technique, which isolates the available design features based on iterative customer specifications. The altered features are then fed into the generative design model for the next iteration. The experiments conducted in this study show that GenYacht can be used as a complete yacht-hull design system. Its underlying components provide notable benefits to maritime vessel architects and designers over contemporary techniques.

10.1 Future Work

The outcome of this thesis showed how integrating computational geometry with data-driven approaches could result in powerful techniques for intuitive design parameterisation and efficient design spaces, which are explored via computationally low-cost solvers with the integration of human intelligence, thereby mitigating the drawback in typical SSD pipelines used in the maritime industry.

However, our contributions are only the initial steps to igniting the importance of radical and intuitive computational tools. We have witnessed the rise of computational power diving the entire ship design through all three phases; i) concept (preliminary) design, ii) contract (full) design and iii) detail (build) design, via efforts like the HOLISHIP project [222]. However, including data-driven power and computational integration of human intelligence within these pipelines is still a path to pave, which will exploit the power of vast and user-centred design spaces. Therefore, there are still many exciting challenges and research directions which are essential to be explored for significant reforms in this field.

10.1.1 Path to simulation-driven design and manufacturing

The technical contribution of this thesis has exhibited that a radical approach to SDD can result in innovative designs meeting all design requirements. It is imperative that the resulting designs be manufactured and used in the real world. Therefore, simulations should not only drive the *design* phase but the entire ship development cycle, including *manufacturing*, while embedding the proper manufacturing criteria and constraints. This will genuinely increase innovation, reduce costs, and streamline production schedules to stay ahead of the competition. Simulation-driven manufacturing is crucial for ship design, where large-scale physical prototyping is nearly impossible. A design resulting from the SDD may be optimal, but its overlooked manufacturability aspect can drain the resources and increase the computational cost, resulting in a significant burden at such a late stage

in the development process.

In contrast, a simulation-driven design and manufacturing (SDDM) approach will be optimistic to deliver manufacturability insights directly to designers from manufacturing engineers right from the preliminary design stage. It will accelerate market placement and enables the exploration of cost-effective workflows, especially for additive manufacturing. Therefore, in the future, we will work on selecting critical manufacturing criteria and constraints to implant them in parallel to the design constraints.

10.1.2 AI to navigate preliminary design stage

To date, the exploration of design spaces in the maritime industry is highly knowledge-driven, gained through an iterative process of trial and error. Optimisation only means using designers' experience-based intuition to take the best out of two to three feasible solutions. In contrast, the design spaces are composed of infinite design possibilities with many potential solutions. Therefore, efficient exploration of rich space for innovative design is a crucial part of the preliminary design stage. This thesis shows that this stage can extensively benefit from intelligent approaches, such as deep learning-based generative models.

However, the primary limitation of these models in maritime is the unavailability of extensive design data with complete manufacturing detail. In this thesis, we have worked on developing a large dataset of various ship designs and have made it public. In the future, extending these efforts and working on constructing and preserving high-quality databases will be beneficial. This would involve building a solid consortium of maritime partners from industry and academia, along with automatic verification systems that allow database input to be efficiently crowd-sourced. The availability of such a database may encourage new AI algorithms and systems specifically born to support maritime design tasks.

10.1.3 Revisiting ship design theory and practice

One pitfall of the maritime industry for lagging behind compared to other fields like automotive and aeronautical towards radical design ideas and adoption of new tools is mainly due to the ill definition of design theory and decentralised design practices. Ships are large and complex entities used for transportation. The designers do not benefit from a prototype, so they must get the design right first. This aspect requires the design to be practised and approached systematically; however, the research on design theory and practice in this field is almost negligible.

There is a need to improve the design practice so it is efficiently mimicked computationally. It is vital to eliminate the existing loopholes, modernised design practices and, more importantly, understand how naval architects perceive design. This is one of the limitations that most of the tools resulting from the research cannot make it to the industry. Most researchers are mainly focused on high-fidelity physical tools leaving the research on developing new design interfaces neglected. Although some tools exist, there is a need to understand how they are used in the industry. Therefore, in the future, we will also work on developing a framework to understand how naval architects approach towards design and what are the most efficient interfaces to integrate the proposed approaches.

10.1.4 End-to-end system and its adoption to industry

Although some academic scholars from the maritime field have contributed considerably to the modernisation of preliminary ship design techniques, their usage in the industry still needs to be improved. The serviceability and survivability of any new design approach are only possible if the industry is willing to take risks and move towards radical design ideas. Throughout this PhD thesis, we have realised that the maritime sector's hindrance to adopting new design tools is mainly due to the unavailability of end-to-end systems. Therefore, in the future, we will develop a complete end-to-end system considering the entire cycle of ship design with the integration of design and manufacturing constraints. The

new system should adopt the concept of digital prototyping to validate design algorithms and techniques. The new system will be tested and marketed to the industry, which is essential to understand the capabilities and limitations of the new design approaches and to provide insight into the core of the existing design problems in maritime.

Appendix A

Software resulting from the thesis

The software resulting from the work in this thesis is written using Matlab, C++ and Python languages, whereas the code for design creation and parametric modelling developed to create design variations are implemented using Rhino3D's scripting language. All the codes related to the works are well documented and uploaded to the GitHub repository[1].

[1] https://github.com/shahrozkhan66

Appendix B

List of Publications resulting from the thesis

B.1 Journal Publications

1. Khan, S., Kaklis, P., Serani, A., & Diez, M. (2022). Geometric moment-dependent global sensitivity analysis without simulation data: application to ship hull form optimisation. Computer-Aided Design, 103339.

2. Khan, S., Kaklis, P., Serani, A., Diez, M., & Kostas, K. (2022). Shape-supervised dimension reduction: Extracting geometry and physics associated features with geometric moments. Computer-Aided Design, 103327.

3. Khan, S., & Kaklis, P. (2021). From regional sensitivity to intra-sensitivity for parametric analysis of free-form shapes: Application to ship design. Advanced Engineering Informatics, 49, 101314.

4. Khan, S., Gunpinar, E., & Sener, B. (2019). GenYacht: An interactive generative design system for computer-aided yacht hull design. Ocean Engineering, 191, 106462.

B.2 Conference publications

1. Khan, S., Kostas, K., Kaklis, P., Serani, A., & Diez, M. (2023). Bayesian shape optimisation in high dimensional design spaces using isogeometric boundary element analysis. In AIAA SciTech 2023 Forum.

2. Khan, S., Gunpinar, E., Mert Dogan, K., Sener, B., & Kaklis, P. (2022, June). ModiYacht: intelligent CAD tool for parametric, generative, attributive and interactive modelling of yacht hull forms. In SNAME 14th International Marine Design Conference. OnePetro.

3. Khan, S., Kaklis, P., Serani, A., & Diez, M. (2022). Supporting Expensive Physical Models With Geometric Moment Invariants to Accelerate Sensitivity Analysis for Shape Optimisation. In AIAA SciTech 2022 Forum (p. 2093).

4. Khan, S., Serani, A., Diez, M., & Kaklis, P. (2021). Physics-informed feature-to-feature learning for design-space dimensionality reduction in shape optimisation. In AIAA scitech 2021 forum (p. 1235).

B.3 Conference Talks

1. Khan, K., Kaklis P. (2022, March 14-16). Intra-sensitivity: An approach to regional sensitivity analysis for studying local behaviour of parametric sensitivities. In Tenth International Conference on Sensitivity Analysis of Model Output (SAMO), Florida State University, Tallahassee, Florida.

2. Khan, S., Kaklis, P., Serani, A., & Diez, M. (2022, June 5-9). Shape Signature Subspace: A shapewise-supervised dimension-reduction approach for shape optimisation. In 8th European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS), Oslo, Norway.

Appendix B. List of Publications resulting from the thesis

3. Khan, S., Kaklis, P., Serani, A., & Diez, M. (2022, 31 July-5 August). Geometric Structure-Preserving Design Space Dimensionality Reduction. In 15th World Congress on Computational Mechanics (WCCM-XV), Yokohama, Japan.

4. Khan, S., Kaklis, P., Serani, A., & Diez, M. (2022, June 27-29). Shape-supervised Dimension Reduction: Extracting Geometry and Physics Associated Features with Geometric Moments. In Symposium on Solid and Physical Modeling (SPM).

5. Khan, S., & Kaklis, P. (2022, June 13-17). Geometry Functionals as Physics-Informed Accelerators of Shape Optimisation. In GRAPES' Second Doctoral School, Lugano, Switzerland.

6. Khan, S., & Kaklis, P. (2022, June 15-16). Intra-sensitivity: Analysing local behaviour of parametric sensitivities in hull form optimisation. In Ocean Energy and Marine Transport Research Conference 2022, Glasgow, United Kingdom.

7. Khan, S. (2022, July 2-6). Shape-supervised dimension reduction. In Deep Learning and Design Workshop - Tenth International Conference on Design Computing and Cognition, Glasgow, United Kingdom.

8. Khan, S. (2022, July 2-6). ModiYacht: Intelligent CAD tool for parametric, generative, attributive and interactive modelling of yacht hull forms. In Design Descriptions: Disruptive Technologies in Future Design & Development Systems Workshop - Tenth International Conference on Design Computing and Cognition, Glasgow, United Kingdom.

9. Khan, S., Kostas, K., Kaklis, P., Serani, A., & Diez, M. (2021, September). Bayesian shape optimization in high dimensional design spaces using IGA-enabled solvers. In Virtual International Conference on Isogeometric Analysis.

10. Khan, S., Kaklis, P., Serani, A., & Diez, M. (2021). Geometry-driven parametric sensitivity analysis for free-form marine shapes. In International Center for Numerical

Methods in Engineering (CIMNE).

11. Kaklis, P., Khan, S., Serani, A., & Diez, M. (2021, November). Supporting expensive physical models with geometric moment invariants to accelerate sensitivity analysis for shape optimisation. In Dagstuhl Seminar 21471: Geometric Modeling: Interoperability and New Challenges.

12. Khan, S. (2021, November 3-4). Geometry-driven parametric sensitivity analysis for free-form marine shapes. In International Conference on Postgraduate Research in Maritime Technology (PostGradMarTec), virtual.

13. Khan, S., & Kaklis, P. (2019, June 2019). Interactive Design System for User-Centered Design of Yacht Hulls. In Doctoral School Multidisciplinary Symposium (DSMS'19), Glasgow, United Kingdom.

14. Khan, S., & Kaklis, P. (2019, June). Intelligent Design Systems for Maritime Industry. In Postgraduate Research Conference (PGR'19), Glasgow, United Kingdom.

Bibliography

[1] K. Kostas, A. Ginnis, C. Politis, and P. Kaklis, "Ship-hull shape optimization with a T-spline based BEM–isogeometric solver," *Computer Methods in Applied Mechanics and Engineering*, vol. 284, pp. 611–622, 2015.

[2] T. Katsoulis, X. Wang, and P. Kaklis, "A T-splines-based parametric modeller for computer-aided ship design," *Ocean Engineering*, vol. 191, p. 106433, 2019.

[3] P. Cignoni, C. Rocchini, and R. Scopigno, "Metro: Measuring error on simplified surfaces," in *Computer graphics forum*, vol. 17. Wiley Online Library, 1998, pp. 167–174.

[4] S. Khan, E. Gunpinar, and K. M. Dogan, "A novel design framework for generation and parametric modification of yacht hull surfaces," *Ocean Engineering*, vol. 136, pp. 243–259, 2017.

[5] E. Gunpinar and S. Khan, "A multi-criteria based selection method using non-dominated sorting for genetic algorithm based design," *Optimization and Engineering*, vol. 21, no. 4, pp. 1319–1357, 2020.

[6] W. Chen, M. Fuge, and J. Chazan, "Design manifolds capture the intrinsic complexity and dimension of design spaces," *Journal of Mechanical Design*, vol. 139, no. 5, 2017.

[7] M. Turrin, P. Von Buelow, and R. Stouffs, "Design explorations of performance driven geometry in architectural design using parametric modeling and genetic algorithms," *Advanced Engineering Informatics*, vol. 25, no. 4, pp. 656–675, 2011.

[8] S. Khan, E. Gunpinar, M. Moriguchi, and H. Suzuki, "Evolving a psycho-physical distance metric for generative design exploration of diverse shapes," *Journal of Mechanical Design*, vol. 141, no. 11, 2019.

[9] A. Schulz, J. Xu, B. Zhu, C. Zheng, E. Grinspun, and W. Matusik, "Interactive design space exploration and optimization for CAD models," *ACM Transactions on Graphics (TOG)*, vol. 36, no. 4, pp. 1–14, 2017.

[10] L. Sun, H. Gao, S. Pan, and J.-X. Wang, "Surrogate modeling for fluid flows based on physics-constrained deep learning without simulation data," *Computer Methods in Applied Mechanics and Engineering*, vol. 361, p. 112732, 2020.

[11] A. Serani, M. Diez, F. van Walree, and F. Stern, "URANS analysis of a free-running destroyer sailing in irregular stern-quartering waves at sea state 7," *Ocean Engineering*, vol. 237, p. 109600, 2021.

[12] D. D'Agostino, A. Serani, and M. Diez, "Design-space assessment and dimensionality reduction: An off-line method for shape reparameterization in simulation-based optimization," *Ocean Engineering*, vol. 197, p. 106852, 2020.

[13] S. Khan and M. J. Awan, "A generative design technique for exploring shape variations," *Advanced Engineering Informatics*, vol. 38, pp. 712–724, 2018.

[14] A. Taber, G. Kumar, M. Freytag, and V. Shapiro, "A moment-vector approach to interoperable analysis," *Computer-Aided Design*, vol. 102, pp. 139–147, 2018.

[15] K. Belibassakis, T. P. Gerostathis, K. Kostas, C. Politis, P. Kaklis, A. Ginnis, and C. Feurer, "A BEM-isogeometric method for the ship wave-resistance problem," *Ocean Engineering*, vol. 60, pp. 53–67, 2013.

[16] Z. Masood, S. Khan, and L. Qian, "Machine learning-based surrogate model for accelerating simulation-driven optimisation of hydropower kaplan turbine," *Renewable Energy*, vol. 173, pp. 827–848, 2021.

[17] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *Journal of Computational Physics*, vol. 378, pp. 686–707, 2019.

[18] S. Khan and P. Kaklis, "From regional sensitivity to intra-sensitivity for parametric analysis of free-form shapes: Application to ship design," *Advanced Engineering Informatics*, vol. 49, p. 101314, 2021.

[19] T. Rios, B. Sendhoff, S. Menzel, T. Bäck, and B. van Stein, "On the efficiency of a point cloud autoencoder as a geometric representation for shape optimization," in *2019 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 2019, pp. 791–798.

[20] T. W. Lukaczyk, P. Constantine, F. Palacios, and J. J. Alonso, "Active subspaces for shape optimization," in *10th AIAA multidisciplinary design optimization conference*, 2014, p. 1171.

[21] S. Khan, A. Serani, M. Diez, and P. Kaklis, "Physics-informed feature-to-feature learning for design-space dimensionality reduction in shape optimisation," in *AIAA Scitech 2021 Forum*, 2021, p. 1235.

[22] M. Diez, E. F. Campana, and F. Stern, "Design-space dimensionality reduction in shape optimization by karhunen–loève expansion," *Computer Methods in Applied Mechanics and Engineering*, vol. 283, pp. 1525–1544, 2015.

[23] D. D'Agostino, A. Serani, E. F. Campana, and M. Diez, "Nonlinear methods for design-space dimensionality reduction in shape optimization," in *International Work-*

shop on Machine Learning, Optimization, and Big Data. Springer, 2017, pp. 121–132.

[24] J. B. Tenenbaum, V. d. Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *science*, vol. 290, no. 5500, pp. 2319–2323, 2000.

[25] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[26] K. Yonekura and K. Suzuki, "Data-driven design exploration method using conditional variational autoencoder for airfoil design," *Structural and Multidisciplinary Optimization*, vol. 64, no. 2, pp. 613–624, 2021.

[27] D. Shu, J. Cunningham, G. Stump, S. W. Miller, M. A. Yukish, T. W. Simpson, and C. S. Tucker, "3D design using generative adversarial networks and physics-based validation," *Journal of Mechanical Design*, vol. 142, no. 7, p. 071701, 2020.

[28] J. Li and M. Zhang, "On deep-learning-based geometric filtering in aerodynamic shape optimization," *Aerospace Science and Technology*, vol. 112, p. 106603, 2021.

[29] C. Çelik, D. B. Danışman, S. Khan, and P. Kaklis, "A reduced order data-driven method for resistance prediction and shape optimization of hull vane," *Ocean Engineering*, vol. 235, p. 109406, 2021.

[30] J. Hamel, M. Li, and S. Azarm, "Design improvement by sensitivity analysis under interval uncertainty using multi-objective optimization," *Journal of mechanical design*, vol. 132, no. 8, 2010.

[31] R. S. Hutcheson and D. A. McAdams, "A hybrid sensitivity analysis for use in early design," *Journal of Mechanical design*, vol. 132, no. 11, 2010.

[32] W. Chen, K. Chiu, and M. D. Fuge, "Airfoil design parameterization and optimization using bézier generative adversarial networks," *AIAA journal*, vol. 58, no. 11, pp. 4723–4735, 2020.

[33] A. Serani, F. Stern, E. F. Campana, and M. Diez, "Hull-form stochastic optimization via computational-cost reduction methods," *Engineering with Computers*, pp. 1–25, 2021.

[34] T. R. Langerak, "Local parameterization of free-form shapes using free-form feature recognition," *Computer-Aided Design*, vol. 42, no. 8, pp. 682–692, 2010.

[35] S. Krish, "A practical generative design method," *Computer-Aided Design*, vol. 43, no. 1, pp. 88–100, 2011.

[36] S. Tarantola, V. Kopustinskas, R. Bolado-Lavin, A. Kaliatka, E. Ušpuras, and M. Vaišnoras, "Sensitivity analysis using contribution to sample variance plot: Application to a water hammer model," *Reliability Engineering & System Safety*, vol. 99, pp. 62–73, 2012.

[37] R. Sheikholeslami, S. Razavi, H. V. Gupta, W. Becker, and A. Haghnegahdar, "Global sensitivity analysis for high-dimensional problems: How to objectively group factors and measure robustness and convergence while reducing computational cost," *Environmental Modelling & Software*, vol. 111, pp. 282–299, 2019.

[38] S. Khan and E. Gunpinar, "Sampling CAD models via an extended teaching–learning-based optimization technique," *Computer-Aided Design*, vol. 100, pp. 52–67, 2018.

[39] M. Yin, X. Zheng, J. D. Humphrey, and G. E. Karniadakis, "Non-invasive inference of thrombus material properties with physics-informed neural networks," *Computer Methods in Applied Mechanics and Engineering*, vol. 375, p. 113603, 2021.

[40] E. Haghighat, A. C. Bekar, E. Madenci, and R. Juanes, "A nonlocal physics-informed deep learning framework using the peridynamic differential operator," *Computer Methods in Applied Mechanics and Engineering*, vol. 385, p. 114012, 2021.

[41] K. M. Dogan and E. Gunpinar, "Learning yacht hull adjectives and their relationship with hull surface geometry using gmdh-type neural networks for human oriented smart design," *Ocean Engineering*, vol. 145, pp. 215–229, 2017.

[42] S. Khan, "Development of a cad system for parametric and attribute-based modification of yacht hull models," Master's thesis, Istanbul Technical University, 2017.

[43] T. P. McDonald, D. J. Andrews, and R. G. Pawling, "A demonstration of an advanced library based approach to the initial design exploration of different hullform configurations," *Computer-Aided Design*, vol. 44, no. 3, pp. 209–223, 2012.

[44] R. Pawling and D. Andrews, "Design sketching for computer aided preliminary ship design," *Ship Technology Research*, vol. 58, no. 3, pp. 182–194, 2011.

[45] E. Duchateau, "Interactive evolutionary concept exploration in preliminary ship design," Ph.D. dissertation, Delft University of Technology, 2016.

[46] B. J. Van Oers, "A packing approach for the early stage design of service vessels," Ph.D. dissertation, Delft University of Technology, 2011.

[47] T. Van Bruinessen, H. Hopman, T. DeNucci, and B. Van Oers, "Generating more valid designs during design exploration," *Journal of Ship Production and Design*, vol. 27, no. 4, pp. 153–161, 2011.

[48] B. Hasubek and S. Harries, "Simulation-driven design of sailing yachts and motor boats," in *Proceedings of the VII International Conference on Computational Methods in Marine Engineering*. International Center for Numerical Methods in Engineering, 2017.

[49] A. Papanikolaou, "Holistic ship design optimization," *Computer-Aided Design*, vol. 42, no. 11, pp. 1028–1044, 2010.

[50] H. Cui, O. Turan, and P. Sayer, "Learning-based ship design optimization approach," *Computer-Aided Design*, vol. 44, no. 3, pp. 186–195, 2012.

[51] A. Krishnamurthy and S. McMains, "Accurate gpu-accelerated surface integrals for moment computation," *Computer-Aided Design*, vol. 43, no. 10, pp. 1284–1295, 2011.

[52] S. Khan, P. Kaklis, A. Serani, and M. Diez, "Geometric moment-dependent global sensitivity analysis without simulation data: application to ship hull form optimisation," *Computer-Aided Design*, p. 103339, 2022.

[53] S. Khan, P. Kaklis, A. Serani, M. Diez, and K. Kostas, "Shape-supervised dimension reduction: Extracting geometry and physics associated features with geometric moments," *Computer-Aided Design*, vol. 150, p. 103327, 2022.

[54] S. Khan, E. Gunpinar, and B. Sener, "Genyacht: An interactive generative design system for computer-aided yacht hull design," *Ocean Engineering*, vol. 191, p. 106462, 2019.

[55] C.-G. Yin and Y.-S. Ma, "Parametric feature constraint modelling and mapping in product development," *Advanced Engineering Informatics*, vol. 26, no. 3, pp. 539–552, 2012.

[56] S. Myung and S. Han, "Knowledge-based parametric design of mechanical products based on configuration design method," *Expert Systems with applications*, vol. 21, no. 2, pp. 99–107, 2001.

[57] J. Cagan, M. I. Campbell, S. Finger, and T. Tomiyama, "A framework for computational design synthesis: Model and applications," *Journal of Computing and Information Science in Engineering*, vol. 5, no. 3, pp. 171–181, 2005.

[58] W. Pan, X. Chen, and S. Gao, "Automatic shape adaptation for parametric solid models," *Computer-Aided Design*, vol. 62, pp. 78–97, 2015.

[59] N. Emami, "Untangling parameters: A formalized framework for identifying overlapping design parameters between two disciplines for creating an interdisciplinary parametric model," *Advanced Engineering Informatics*, vol. 42, p. 100943, 2019.

[60] J. A. Samareh, "A survey of shape parameterization techniques," in *CEAS AIAA ICASE NASA Langley International Forum on Aeroelasticity and Structural Dynamics*, 1999, pp. 333–343.

[61] T. W. Sederberg and S. R. Parry, "Free-form deformation of solid geometric models," in *Proceedings of the 13th annual conference on Computer graphics and interactive techniques*. Association for Computing Machinery, 1986, pp. 151–160.

[62] J. Kouh and S. Chau, "Computer-aided geometric design and panel generation for hull forms based on rational cubic bezier curves," *Computer aided geometric design*, vol. 10, no. 6, pp. 537–549, 1993.

[63] F. Perez, J. Clemente, J. Suárez, and J. González, "Parametric generation, modeling, and fairing of simple hull lines with the use of nonuniform rational b-spline surfaces," *Journal of Ship Research*, vol. 52, no. 1, pp. 1–15, 2008.

[64] A. Rodríguez and L. Fernández-Jambrina, "Programmed design of ship forms," *Computer-Aided Design*, vol. 44, no. 7, pp. 687–696, 2012.

[65] H. Lackenby, "On the systematic geometrical variation of ship forms," *Transactions of the Royal Institute of Naval Architects (RINA)*, vol. 92, pp. 289–315, 1950.

[66] S. Menzel and B. Sendhoff, "Representing the change-free form deformation for evolutionary design optimisation," in *Evolutionary computation in practice*. Springer, 2008, pp. 63–86.

[67] W. Chen and M. Fuge, "Beyond the known: Detecting novel feasible domains over an unbounded design space," *Journal of Mechanical Design*, vol. 139, no. 11, 2017.

[68] D. Wu and G. Gary Wang, "Knowledge-assisted optimization for large-scale design problems: A review and proposition," *Journal of Mechanical Design*, vol. 142, no. 1, 2020.

[69] K. M. Dogan, H. Suzuki, E. Gunpinar, and M.-S. Kim, "A generative sampling system for profile designs with shape constraints and user evaluation," *Computer-Aided Design*, vol. 111, pp. 93–112, 2019.

[70] M. M. Singh and P. Geyer, "Information requirements for multi-level-of-development BIM using sensitivity analysis for energy performance," *Advanced Engineering Informatics*, vol. 43, p. 101026, 2020.

[71] X. Cheng, G. Li, R. Skulstad, P. Major, S. Chen, H. P. Hildre, and H. Zhang, "Data-driven uncertainty and sensitivity analysis for ship motion modeling in offshore operations," *Ocean Engineering*, vol. 179, pp. 261–272, 2019.

[72] F. Pianosi, K. Beven, J. Freer, J. W. Hall, J. Rougier, D. B. Stephenson, and T. Wagener, "Sensitivity analysis of environmental models: A systematic review with practical workflow," *Environmental Modelling & Software*, vol. 79, pp. 214–232, 2016.

[73] L. Pronzato, "Sensitivity analysis via Karhunen–Loève expansion of a random field model: Estimation of Sobol' indices and experimental design," *Reliability Engineering & System Safety*, vol. 187, pp. 93–109, 2019.

[74] R. Sheikholeslami and S. Razavi, "Progressive latin hypercube sampling: An efficient approach for robust sampling-based analysis of environmental models," *Environmental modelling & software*, vol. 93, pp. 109–126, 2017.

[75] J. Morio, "Global and local sensitivity analysis methods for a physical system," *European journal of physics*, vol. 32, no. 6, p. 1577, 2011.

[76] M. Fesanghary, E. Damangir, and I. Soleimani, "Design optimization of shell and tube heat exchangers using global sensitivity analysis and harmony search algorithm," *Applied Thermal Engineering*, vol. 29, no. 5-6, pp. 1026–1031, 2009.

[77] E. Borgonovo and E. Plischke, "Sensitivity analysis: A review of recent advances," *European Journal of Operational Research*, vol. 248, no. 3, pp. 869–887, 2016.

[78] A. Saltelli, M. Ratto, T. Andres, F. Campolongo, J. Cariboni, D. Gatelli, M. Saisana, and S. Tarantola, *Global sensitivity analysis: The primer*. John Wiley & Sons, 2008.

[79] P. G. Constantine, *Active subspaces: Emerging ideas for dimension reduction in parameter studies*. SIAM, 2015, vol. 2.

[80] P. G. Constantine and P. Diaz, "Global sensitivity metrics from active subspaces," *Reliability Engineering & System Safety*, vol. 162, pp. 1–13, 2017.

[81] P. G. Constantine, M. Emory, J. Larsson, and G. Iaccarino, "Exploiting active subspaces to quantify uncertainty in the numerical simulation of the HyShot II scramjet," *Journal of Computational Physics*, vol. 302, pp. 1–20, 2015.

[82] Z. Jiang and J. Li, "High dimensional structural reliability with dimension reduction," *Structural Safety*, vol. 69, pp. 35–46, 2017.

[83] M. Tezzele, F. Salmoiraghi, A. Mola, and G. Rozza, "Dimension reduction in heterogeneous parametric spaces with application to naval engineering shape design problems," *Advanced Modeling and Simulation in Engineering Sciences*, vol. 5, no. 1, pp. 1–19, 2018.

[84] J. Sinclair, "Response to the PSACOIN level S exercise. PSACOIN level S intercomparison," *Nuclear Energy Agency, OECDParis, France*, 1993.

[85] R. Bolado-Lavin, W. Castaings, and S. Tarantola, "Contribution to the sample mean plot for graphical and numerical sensitivity analysis," *Reliability Engineering & System Safety*, vol. 94, no. 6, pp. 1041–1049, 2009.

[86] P. Wei, Z. Lu, W. Ruan, and J. Song, "Regional sensitivity analysis using revised mean and variance ratio functions," *Reliability Engineering & System Safety*, vol. 121, pp. 121–135, 2014.

[87] P. Wei, Z. Lu, and J. Song, "Regional and parametric sensitivity analysis of Sobol' indices," *Reliability Engineering & System Safety*, vol. 137, pp. 87–100, 2015.

[88] J. Wu, "A new sequential space-filling sampling strategy for elementary effects-based screening method," *Applied Mathematical Modelling*, vol. 83, pp. 419–437, 2020.

[89] W. Gong, Q. Duan, J. Li, C. Wang, Z. Di, A. Ye, C. Miao, and Y. Dai, "An intercomparison of sampling methods for uncertainty quantification of environmental dynamic models," *J. Environ. Inf*, vol. 28, no. 1, pp. 11–24, 2016.

[90] C. B. Storlie, L. P. Swiler, J. C. Helton, and C. J. Sallaberry, "Implementation and evaluation of nonparametric regression procedures for sensitivity analysis of computationally demanding models," *Reliability Engineering & System Safety*, vol. 94, no. 11, pp. 1735–1763, 2009.

[91] G. Blatman and B. Sudret, "Efficient computation of global sensitivity indices using sparse polynomial chaos expansions," *Reliability Engineering & System Safety*, vol. 95, no. 11, pp. 1216–1229, 2010.

[92] K. Cheng and Z. Lu, "Adaptive sparse polynomial chaos expansions for global sensitivity analysis based on support vector regression," *Computers & Structures*, vol. 194, pp. 86–96, 2018.

[93] K. Konakli and B. Sudret, "Global sensitivity analysis using low-rank tensor approximations," *Reliability Engineering & System Safety*, vol. 156, pp. 64–83, 2016.

[94] M. Li, R.-Q. Wang, and G. Jia, "Efficient dimension reduction and surrogate-based sensitivity analysis for expensive models with high-dimensional outputs," *Reliability Engineering & System Safety*, vol. 195, p. 106725, 2020.

[95] J. Song, P. Wei, M. A. Valdebenito, M. Faes, and M. Beer, "Data-driven and active learning of variance-based sensitivity indices with bayesian probabilistic integration," *Mechanical Systems and Signal Processing*, vol. 163, p. 108106, 2022.

[96] W. Chen and F. Ahmed, "PaDGAN: Learning to generate high-quality novel designs," *Journal of Mechanical Design*, vol. 143, no. 3, 2021.

[97] P. Constantine and D. Gleich, "Computing active subspaces with Monte Carlo," *arXiv preprint arXiv:1408.0545*, 2014.

[98] C. Othmer, T. W. Lukaczyk, P. Constantine, and J. J. Alonso, "On active subspaces in car aerodynamics," in *17th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*. American Institute of Aeronautics and Astronautics, 2016, p. 4294.

[99] S. J. Bates, J. Sienz, and D. S. Langley, "Formulation of the Audze–Eglais uniform latin hypercube design of experiments," *Advances in Engineering Software*, vol. 34, no. 8, pp. 493–506, 2003.

[100] T. W. Sederberg, D. L. Cardon, G. T. Finnigan, N. S. North, J. Zheng, and T. Lyche, "T-spline simplification and local refinement," *ACM transactions on graphics (TOG)*, vol. 23, no. 3, pp. 276–283, 2004.

[101] A. Bressan, "Some properties of LR-splines," *Computer Aided Geometric Design*, vol. 30, no. 8, pp. 778–794, 2013.

[102] C. Giannelli, B. Jüttler, and H. Speleers, "Thb-splines: The truncated basis for hierarchical splines," *Computer Aided Geometric Design*, vol. 29, no. 7, pp. 485–498, 2012.

[103] S. H. Greshake and R. Bronsart, "Application of subdivision surfaces in ship hull form modeling," *Computer-Aided Design*, vol. 100, pp. 79–92, 2018.

[104] M. E. Yumer, P. Asente, R. Mech, and L. B. Kara, "Procedural modeling using autoencoder networks," in *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, 2015, pp. 109–118.

[105] J. Huang, L. Chen, X. Liu, and H. Bao, "Efficient mesh deformation using tetrahedron control mesh," in *Proceedings of the 2008 ACM symposium on Solid and physical modeling*, 2008, pp. 241–247.

[106] A. Coppedé, G. Vernengo, and D. Villa, "A combined approach based on subdivision surface and free-form deformation for smart ship hull form design and variation," *Ships and Offshore Structures*, vol. 13, no. 7, pp. 769–778, 2018.

[107] A. Krishnamurthy, S. McMains, and I. Hanniel, "GPU-accelerated Hausdorff distance computation between dynamic deformable nurbs surfaces," *Computer-Aided Design*, vol. 43, no. 11, pp. 1370–1379, 2011.

[108] J. Holtrop and G. Mennen, "An approximate power prediction method," *International Shipbuilding Progress*, vol. 29, no. 335, pp. 166–170, 1982.

[109] R. Rao, "Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems," *International Journal of Industrial Engineering Computations*, vol. 7, no. 1, pp. 19–34, 2016.

[110] S. Chouliaras, P. Kaklis, K. Kostas, A. Ginnis, and C. Politis, "An isogeometric boundary element method for 3D lifting flows using T-splines," *Computer Methods in Applied Mechanics and Engineering*, vol. 373, p. 113556, 2021.

[111] K. V. Kostas, A. I. Ginnis, C. G. Politis, and P. D. Kaklis, "Shape-optimization of 2D hydrofoils using an isogeometric BEM solver," *Computer-Aided Design*, vol. 82, pp. 79–87, 2017.

[112] C. Celik, D. Danisman, P. Kaklis, and S. Khan, "An investigation into the effect of the hull vane on the ship resistance in openfoam," in *Sustainable Development and*

Innovations in Marine Technologies: Proceedings of the 18th International Congress of the Maritime Association of the Mediterranean (IMAM 2019), 2019, p. 136.

[113] P. Jin, B. Xie, and F. Xiao, "Multi-moment finite volume method for incompressible flows on unstructured moving grids and its application to fluid-rigid body interactions," *Computers & Structures*, vol. 221, pp. 91–110, 2019.

[114] G. Kumar and A. Taber, "An integral representation of fields with applications to finite element analysis of spatially varying materials," *Computer-Aided Design*, vol. 126, p. 102869, 2020.

[115] B. Müller, F. Kummer, and M. Oberlack, "Highly accurate surface and volume integration on implicit domains by means of moment-fitting," *International Journal for Numerical Methods in Engineering*, vol. 96, no. 8, pp. 512–528, 2013.

[116] A. M. Bronstein, M. M. Bronstein, and R. Kimmel, *Numerical geometry of non-rigid shapes*. Springer Science & Business Media, 2008.

[117] L. Luciano and A. B. Hamza, "A global geometric framework for 3d shape retrieval using deep learning," *Computers & Graphics*, vol. 79, pp. 14–23, 2019.

[118] D. F. Atrevi, D. Vivet, F. Duculty, and B. Emile, "A very simple framework for 3d human poses estimation using a single 2d image: Comparison of geometric moments descriptors," *Pattern Recognition*, vol. 71, pp. 389–401, 2017.

[119] O. Klepper, "Multivariate aspects of model uncertainty analysis: tools for sensitivity analysis and calibration," *Ecological modelling*, vol. 101, no. 1, pp. 1–13, 1997.

[120] P. Bassanini, U. Bulgarelli, E. F. Campana, and F. Lalli, "The wave resistance problem in a boundary integral formulation," *Surveys on Mathematics for Industry*, vol. 4, pp. 151–194, 1994.

[121] L. Yang, F. Albregtsen, and T. Taxt, "Fast computation of three-dimensional geometric moments using a discrete divergence theorem and a generalization to higher dimensions," *Graphical models and image processing*, vol. 59, no. 2, pp. 97–108, 1997.

[122] F. Gamboa, A. Janon, T. Klein, and A. Lagnoux, "Sensitivity indices for multivariate outputs," *Comptes Rendus Mathematique*, vol. 351, no. 7, pp. 307–310, 2013.

[123] A. Rajan, F. J. Luo, Y. C. Kuang, Y. Bai, and M. P.-L. Ooi, "Reliability-based design optimisation of structural systems using high-order analytical moments," *Structural Safety*, vol. 86, p. 101970, 2020.

[124] P. Milanfar, M. Putinar, J. Varah, B. Gustafsson, and G. H. Golub, "Shape reconstruction from moments: theory, algorithms, and applications," in *Advanced Signal Processing Algorithms, Architectures, and Implementations X*, vol. 4116. International Society for Optics and Photonics, 2000, pp. 406–416.

[125] H.-G. Bui, D. Schillinger, and G. Meschke, "Efficient cut-cell quadrature based on moment fitting for materially nonlinear analysis," *Computer Methods in Applied Mechanics and Engineering*, vol. 366, p. 113050, 2020.

[126] Z. Zhang, W. Jiang, J. E. Dolbow, and B. W. Spencer, "A modified moment-fitted integration scheme for X-FEM applications with history-dependent material data," *Computational Mechanics*, vol. 62, no. 2, pp. 233–252, 2018.

[127] V. Thiagarajan and V. Shapiro, "Shape aware quadratures," *Journal of Computational Physics*, vol. 374, pp. 1239–1260, 2018.

[128] C. Hafner, C. Schumacher, E. Knoop, T. Auzinger, B. Bickel, and M. Bächer, "X-CAD: optimizing cad models with extended finite elements," *ACM Transactions on Graphics (TOG)*, vol. 38, no. 6, pp. 1–15, 2019.

[129] B. Gustafsson, C. He, P. Milanfar, and M. Putinar, "Reconstructing planar domains from their moments," *Inverse Problems*, vol. 16, no. 4, p. 1053, 2000.

[130] A. Kousholt and J. Schulte, "Reconstruction of convex bodies from moments," *Discrete & Computational Geometry*, vol. 65, no. 1, pp. 1–42, 2021.

[131] S. A. Sheynin and A. V. Tuzikov, "Explicit formulae for polyhedra moments," *Pattern Recognition Letters*, vol. 22, no. 10, pp. 1103–1109, 2001.

[132] D. Xu and H. Li, "Geometric moment invariants," *Pattern recognition*, vol. 41, no. 1, pp. 240–249, 2008.

[133] I. M. Sobol, "Global sensitivity indices for nonlinear mathematical models and their monte carlo estimates," *Mathematics and computers in simulation*, vol. 55, no. 1-3, pp. 271–280, 2001.

[134] A. Saltelli, P. Annoni, I. Azzini, F. Campolongo, M. Ratto, and S. Tarantola, "Variance based sensitivity analysis of model output. design and estimator for the total sensitivity index," *Computer physics communications*, vol. 181, no. 2, pp. 259–270, 2010.

[135] S. Kucherenko, B. Feil, N. Shah, and W. Mauntz, "The identification of model effective dimensions using global sensitivity analysis," *Reliability Engineering & System Safety*, vol. 96, no. 4, pp. 440–449, 2011.

[136] K. Campbell, M. D. McKay, and B. J. Williams, "Sensitivity analysis when model outputs are functions," *Reliability Engineering & System Safety*, vol. 91, no. 10-11, pp. 1468–1472, 2006.

[137] M. Lamboni, H. Monod, and D. Makowski, "Multivariate sensitivity analysis to measure global contribution of input factors in dynamic models," *Reliability Engineering & System Safety*, vol. 96, no. 4, pp. 450–459, 2011.

[138] O. Garcia-Cabrejo and A. Valocchi, "Global sensitivity analysis for multivariate output using polynomial chaos expansion," *Reliability Engineering & System Safety*, vol. 126, pp. 25–36, 2014.

[139] X.-Y. Zhang, M. N. Trame, L. J. Lesko, and S. Schmidt, "Sobol sensitivity analysis: a tool to guide the development and evaluation of systems pharmacology models," *CPT: pharmacometrics & systems pharmacology*, vol. 4, no. 2, pp. 69–79, 2015.

[140] S. Han, Y.-S. Lee, and Y. B. Choi, "Hydrodynamic hull form optimization using parametric models," *Journal of marine science and technology*, vol. 17, no. 1, pp. 1–17, 2012.

[141] H.-C. Kim, "On the volumetric balanced variation of ship forms," *Journal of Ocean Engineering and Technology*, vol. 27, no. 2, pp. 1–7, 2013.

[142] C. Abt and S. Harries, "A new approach to integration of cad and cfd for naval architects," in *Sixth international conference on computer applications and information technology in the maritime industries (COMPIT), Cortona*, 2007, pp. 467–479.

[143] L. Birk, WEGEMT., and W. S. School, *Optimistic: optimization in marine design*. Mensch & Buch Verlag., 2003.

[144] E. O. Tuck, "Shallow-water flows past slender bodies," *Journal of fluid mechanics*, vol. 26, no. 1, pp. 81–95, 1966.

[145] ——, "Wave resistance of thin ships and catamarans," *Applied Mathematics Report T8701*, 1987.

[146] J. V. Wehausen, "The wave resistance of ships," in *Advances in applied mechanics*. Elsevier, 1973, vol. 13, pp. 93–245.

[147] T. J. Hughes, J. A. Cottrell, and Y. Bazilevs, "Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement," *Computer methods in applied mechanics and engineering*, vol. 194, no. 39-41, pp. 4135–4195, 2005.

[148] R. O. Fox, F. Laurent, and A. Vié, "Conditional hyperbolic quadrature method of moments for kinetic equations," *Journal of Computational Physics*, vol. 365, pp. 269–293, 2018.

[149] E. Kalogerakis, M. Averkiou, S. Maji, and S. Chaudhuri, "3D shape segmentation with projective convolutional networks," in *proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 3779–3788.

[150] S. Khan, P. Kaklis, A. Serani, and M. Diez, "Supporting expensive physical models with geometric moment invariants to accelerate sensitivity analysis for shape optimisation," in *AIAA SCITECH 2022 Forum*. American Institute of Aeronautics and Astronautics, 2022, p. 2093.

[151] K. Kostas, A. Amiralin, S. Sagimbayev, T. Massalov, Y. Kalel, and C. Politis, "Parametric model for the reconstruction and representation of hydrofoils and airfoils," *Ocean Engineering*, vol. 199, p. 107020, 2020.

[152] E. Gunpinar, U. C. Coskun, M. Ozsipahi, and S. Gunpinar, "A generative design and drag coefficient prediction system for sedan car side silhouettes based on computational fluid dynamics," *Computer-Aided Design*, vol. 111, pp. 65–79, 2019.

[153] A. Burnap, Y. Pan, Y. Liu, Y. Ren, H. Lee, R. Gonzalez, and P. Y. Papalambros, "Improving design preference prediction accuracy using feature learning," *Journal of Mechanical Design*, vol. 138, no. 7, 2016.

[154] A. Serani and M. Diez, "Shape optimization under stochastic conditions by design-space augmented dimensionality reduction," in *19th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference (MA&O), AVIATION 2018*, Atlanta, GA, USA, June 25-29, 2018.

[155] A. Serani, D. D'Agostino, E. F. Campana, and M. Diez, "Assessing the interplay of shape and physical parameters by unsupervised nonlinear dimensionality reduction methods," *Journal of Ship Research*, 2019.

[156] C. K. Williams and C. E. Rasmussen, *Gaussian processes for machine learning*. MIT press Cambridge, MA, 2006, vol. 2, no. 3.

[157] E. Schulz, M. Speekenbrink, and A. Krause, "A tutorial on gaussian process regression: Modelling, exploring, and exploiting functions," *Journal of Mathematical Psychology*, vol. 85, pp. 1–16, 2018.

[158] N. Umetani and B. Bickel, "Learning three-dimensional flow for interactive aerodynamic design," *ACM Transactions on Graphics (TOG)*, vol. 37, no. 4, pp. 1–10, 2018.

[159] J. Snoek, H. Larochelle, and R. P. Adams, "Practical bayesian optimization of machine learning algorithms," in *Advances in neural information processing systems*, 2012, pp. 2951–2959.

[160] M. Pastor, M. Binda, and T. Harčarik, "Modal assurance criterion," *Procedia Engineering*, vol. 48, pp. 543–548, 2012.

[161] Z. Lodhia, A. Rasool, and G. Hajela, "A survey on machine learning and outlier detection techniques," *International Journal of Computer Science and Network Security*, vol. 17, no. 5, pp. 271–276, 2017.

[162] H. Nowacki, "Five decades of computer-aided ship design," *Computer-Aided Design*, vol. 42, no. 11, pp. 956–969, 2010.

[163] A. Ginnis, K. Kostas, C. Feurer, K. Belibassakis, T. Gerostathis, C. Politis, and P. Kaklis, "A catia®ship-parametric model for isogeometric hull optimization with respect to wave resistance," in *Proceedings of ICCAS 2011 conference, Trieste 20-22 September, Italy*, 2011.

[164] S. Khan, E. Gunpinar, K. Mert Dogan, B. Sener, and P. Kaklis, "Modiyacht: Intelligent cad tool for parametric, generative, attributive and interactive modelling of yacht hull forms," in *SNAME 14th International Marine Design Conference*. OnePetro, 2022.

[165] Y. Yu, Z. Gong, P. Zhong, and J. Shan, "Unsupervised representation learning with deep convolutional neural network for remote sensing images," in *International conference on image and graphics*. Springer, 2017, pp. 97–108.

[166] J. Li, M. Zhang, J. R. Martins, and C. Shu, "Efficient aerodynamic shape optimization with deep-learning-based geometric filtering," *AIAA Journal*, vol. 58, no. 10, pp. 4243–4259, 2020.

[167] I. Goodfellow, "Nips 2016 tutorial: Generative adversarial networks," *arXiv preprint arXiv:1701.00160*, 2016.

[168] S. Oh, Y. Jung, S. Kim, I. Lee, and N. Kang, "Deep generative design: Integration of topology optimization and generative models," *Journal of Mechanical Design*, vol. 141, no. 11, 2019.

[169] Z. Nie, T. Lin, H. Jiang, and L. B. Kara, "Topologygan: Topology optimization using generative adversarial networks based on physical fields over the initial domain," *Journal of Mechanical Design*, vol. 143, no. 3, 2021.

[170] J. Wang, W. W. Chen, D. Da, M. Fuge, and R. Rai, "IH-GAN: A conditional generative model for implicit surface-based inverse design of cellular structures," *Computer Methods in Applied Mechanics and Engineering*, vol. 396, p. 115060, 2022.

[171] A. Heyrani Nobari, M. F. Rashad, and F. Ahmed, "Creativegan: Editing generative adversarial networks for creative design synthesis," in *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, vol. 85383. American Society of Mechanical Engineers, 2021, p. V03AT03A002.

[172] Q. Chen, J. Wang, P. Pope, M. Fuge *et al.*, "Inverse design of two-dimensional airfoils using conditional generative models and surrogate log-likelihoods," *Journal of Mechanical Design*, vol. 144, no. 2, 2022.

[173] A. Heyrani Nobari, W. Chen, and F. Ahmed, "Pcdgan: A continuous conditional diverse generative adversarial network for inverse design," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2021, pp. 606–616.

[174] W. Chen and F. Ahmed, "Mo-padgan: Reparameterizing engineering designs for augmented multi-objective optimization," *Applied Soft Computing*, vol. 113, p. 107909, 2021.

[175] W. Chen and M. Fuge, "Synthesizing designs with interpart dependencies using hierarchical generative adversarial networks," *Journal of Mechanical Design*, vol. 141, no. 11, 2019.

[176] A. H. Nobari, W. Chen, and F. Ahmed, "Range-constrained generative adversarial network: Design synthesis under constraints using conditional generative adversarial networks," *Journal of Mechanical Design*, vol. 144, no. 2, 2022.

[177] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 4401–4410.

[178] S. Radhakrishnan, V. Bharadwaj, V. Manjunath, and R. Srinath, "Creative intelligence–automating car design studio with generative adversarial networks (gan)," in *International Cross-Domain Conference for Machine Learning and Knowledge Extraction*. Springer, 2018, pp. 160–175.

[179] L. Regenwetter, A. H. Nobari, and F. Ahmed, "Deep generative models in engineering design: A review," *Journal of Mechanical Design*, vol. 144, no. 7, p. 071704, 2022.

[180] N. C. Brown and C. T. Mueller, "Quantifying diversity in parametric design: a comparison of possible metrics," *AI EDAM*, vol. 33, no. 1, pp. 40–53, 2019.

[181] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne." *Journal of machine learning research*, vol. 9, no. 11, 2008.

[182] M. Bole, "Interactive hull form transformations using curve network deformation," *Ship Technology Research*, vol. 58, no. 1, pp. 46–64, 2011.

[183] A. Schulz, H. Wang, E. Crinspun, J. Solomon, and W. Matusik, "Interactive exploration of design trade-offs," *ACM Transactions on Graphics (TOG)*, vol. 37, no. 4, p. 131, 2018.

[184] A. M. Brintrup, J. Ramsden, H. Takagi, and A. Tiwari, "Ergonomic chair design by fusing qualitative and quantitative criteria using interactive genetic algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 3, pp. 343–354, 2008.

[185] A. Mortazavi, V. Toğan, and A. Nuhoğlu, "Interactive search algorithm: A new hybrid metaheuristic optimization algorithm," *Engineering Applications of Artificial Intelligence*, vol. 71, pp. 275–292, 2018.

[186] E. Poirson, J.-F. Petiot, L. Boivin, and D. Blumenthal, "Eliciting user perceptions using assessment tests based on an interactive genetic algorithm," *Journal of Mechanical Design*, vol. 135, no. 3, p. 031004, 2013.

[187] J. Felkner, E. Chatzi, and T. Kotnik, "Interactive truss design using particle swarm optimization and nurbs curves," *Journal of Building Engineering*, vol. 4, pp. 60–74, 2015.

[188] R. Dou, C. Zong, and G. Nan, "Multi-stage interactive genetic algorithm for collaborative product customization," *Knowledge-Based Systems*, vol. 92, pp. 43–54, 2016.

[189] S. Chaudhuri, E. Kalogerakis, S. Giguere, and T. Funkhouser, "Attribit: content creation with semantic attributes," in *Proceedings of the 26th annual ACM symposium on User interface software and technology*. ACM, 2013, pp. 193–202.

[190] G. Nishida, I. Garcia-Dorado, D. G. Aliaga, B. Benes, and A. Bousseau, "Interactive sketching of urban procedural models," *ACM Transactions on Graphics (TOG)*, vol. 35, no. 4, p. 130, 2016.

[191] K. Liu, X. Zeng, P. Bruniaux, X. Tao, X. Yao, V. Li, and J. Wang, "3d interactive garment pattern-making technology," *Computer-Aided Design*, vol. 104, pp. 113–124, 2018.

[192] N. Umetani, "Exploring generative 3d shapes using autoencoder networks," in *SIGGRAPH Asia 2017 Technical Briefs*. ACM, 2017, p. 24.

[193] E. Gunpinar, S. E. Ovur, and S. Gunpinar, "A user-centered side silhouette generation system for sedan cars based on shape templates," *Optimization and Engineering*, pp. 1–41, 2018.

[194] S. Wang and H. Takagi, "Improving the performance of predicting users' subjective evaluation characteristics to reduce their fatigue in iec," *Journal of physiological anthropology and applied human science*, vol. 24, no. 1, pp. 81–85, 2005.

[195] R. Dou, C. Zong, and M. Li, "An interactive genetic algorithm with the interval arithmetic based on hesitation and its application to achieve customer collaborative product configuration design," *Applied Soft Computing*, vol. 38, pp. 384–394, 2016.

[196] L. García-Hernández, H. Pierreval, L. Salas-Morera, and A. Arauzo-Azofra, "Handling qualitative aspects in unequal area facility layout problem: An interactive genetic algorithm," *Applied Soft Computing*, vol. 13, no. 4, pp. 1718–1727, 2013.

[197] A. T. Machwe and I. C. Parmee, "Reducing user fatigue within an interactive evolutionary design system using clustering and case-based reasoning," *Engineering Optimization*, vol. 41, no. 9, pp. 871–887, 2009.

[198] Z. Gu, M. X. Tang, and J. H. Frazer, "Capturing aesthetic intention during interactive evolution," *Computer-Aided Design*, vol. 38, no. 3, pp. 224–237, 2006.

[199] O. Bandte, "A broad and narrow approach to interactive evolutionary design—an aircraft design example," *Applied Soft Computing*, vol. 9, no. 1, pp. 448–455, 2009.

[200] A. Ramírez, J. R. Romero, and S. Ventura, "Interactive multi-objective evolutionary optimization of software architectures," *Information Sciences*, vol. 463, pp. 92–109, 2018.

[201] R. A. Danhaive and C. T. Mueller, "Combining parametric modeling and interactive optimization for high-performance and creative structural design," in *Proceedings of IASS Annual Symposia*, vol. 2015. International Association for Shell and Spatial Structures (IASS), 2015, pp. 1–11.

[202] J. Zheng, G. Yu, Q. Zhu, X. Li, and J. Zou, "On decomposition methods in interactive user-preference based optimization," *Applied Soft Computing*, vol. 52, pp. 952–973, 2017.

[203] T. W. DeNucci, "Capturing design: Improving conceptual ship design through the capture of design rationale," Ph.D. dissertation, Delft University of Technology, 2012.

[204] M. Diez and D. Peri, "Robust optimization for ship conceptual design," *Ocean Engineering*, vol. 37, pp. 966–977, 2010.

[205] Autodesk, "Project fractal," 2018. [Online]. Available: https://home.fractal.live/

[206] J. Matejka, M. Glueck, E. Bradner, A. Hashemi, T. Grossman, and G. Fitzmaurice, "Dream lens: Exploration and visualization of large-scale generative design datasets," in *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, 2018, p. 369.

[207] R. H. Kazi, T. Grossman, H. Cheong, A. Hashemi, and G. Fitzmaurice, "Dreamsketch: Early stage 3d design explorations with sketching and generative design," in *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*. ACM, 2017, pp. 401–414.

[208] L. Zaman, W. Stuerzlinger, C. Neugebauer, R. Woodbury, M. Elkhaldi, N. Shireen, and M. Terry, "Gem-ni: A system for creating and managing alternatives in generative design," in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM, 2015, pp. 1201–1210.

[209] L. Zaman, W. Stuerzlinger, and C. Neugebauer, "Mace: A new interface for comparing and editing of multiple alternative documents for generative design," in *Proceedings of the 2017 ACM Symposium on Document Engineering*. ACM, 2017, pp. 67–76.

[210] E. Gunpinar and S. Gunpinar, "A shape sampling technique via particle tracing for cad models," *Graphical Models*, vol. 96, pp. 11–29, 2018.

[211] P. Janssen, "Dexen: A scalable and extensible platform for experimenting with population-based design exploration algorithms," *AI EDAM*, vol. 29, no. 4, pp. 443–455, 2015.

[212] L. Caldas, "Generation of energy-efficient architecture solutions applying gene_arch: An evolution-based generative design system," *Advanced Engineering Informatics*, vol. 22, no. 1, pp. 59–70, 2008.

[213] P. Audze, "New approach to planning out of experiments," *Problems of dynamics and strengths*, vol. 35, pp. 104–107, 1977.

[214] R. L. Haupt, S. E. Haupt, and S. E. Haupt, *Practical genetic algorithms*. Wiley New York, 1998, vol. 2.

[215] J. Kennedy, "Particle swarm optimization," in *Encyclopedia of machine learning*. Springer, 2011, pp. 760–766.

[216] D. Karaboga and B. Basturk, "On the performance of artificial bee colony (abc) algorithm," *Applied soft computing*, vol. 8, no. 1, pp. 687–697, 2008.

[217] R. V. Rao, V. J. Savsani, and D. Vakharia, "Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems," *Computer-Aided Design*, vol. 43, no. 3, pp. 303–315, 2011.

[218] R. V. Rao and G. Waghmare, "A new optimization algorithm for solving complex constrained design optimization problems," *Engineering Optimization*, vol. 49, no. 1, pp. 60–83, 2017.

[219] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *IEEE transactions on evolutionary computation*, vol. 6, no. 2, pp. 182–197, 2002.

[220] J. Keuning and M. Katgert, "A bare hull resistance prediction method derived from the results of the delft systematic yacht hull series extended to higher speeds," in *International Conference on Innovation in High Performance Sailing Yachts, Lorient, France*, 2008.

[221] E. V. Lewis, "Principles of naval architecture second revision," *Jersey: SNAME*, vol. 2, 1988.

[222] A. Papanikolaou, S. Harries, P. Hooijmans, J. Marzi, R. Le Néna, S. Torben, A. Yrjänäinen, and B. Boden, "A holistic approach to ship design: Tools and applications," *Journal of Ship Research*, vol. 66, no. 01, pp. 25–53, 2022.