

---

## Numerical and Parametric Optimization of Ship Hull Form

**Auteur** : Dhanani, Smit Paresh

**Promoteur(s)** : Rigo, Philippe

**Faculté** : Faculté des Sciences appliquées

**Diplôme** : Master : ingénieur civil mécanicien, à finalité spécialisée en "Advanced Ship Design"

**Année académique** : 2023-2024

**URI/URL** : <http://hdl.handle.net/2268.2/22262>

---

### *Avertissement à l'attention des usagers :*

*Tous les documents placés en accès ouvert sur le site le site MatheO sont protégés par le droit d'auteur. Conformément aux principes énoncés par la "Budapest Open Access Initiative"(BOAI, 2002), l'utilisateur du site peut lire, télécharger, copier, transmettre, imprimer, chercher ou faire un lien vers le texte intégral de ces documents, les disséquer pour les indexer, s'en servir de données pour un logiciel, ou s'en servir à toute autre fin légale (ou prévue par la réglementation relative au droit d'auteur). Toute utilisation du document à des fins commerciales est strictement interdite.*

*Par ailleurs, l'utilisateur s'engage à respecter les droits moraux de l'auteur, principalement le droit à l'intégrité de l'oeuvre et le droit de paternité et ce dans toute utilisation que l'utilisateur entreprend. Ainsi, à titre d'exemple, lorsqu'il reproduira un document par extrait ou dans son intégralité, l'utilisateur citera de manière complète les sources telles que mentionnées ci-dessus. Toute utilisation non explicitement autorisée ci-avant (telle que par exemple, la modification du document ou son résumé) nécessite l'autorisation préalable et expresse des auteurs ou de leurs ayants droit.*

---



Universität  
Rostock



Traditio et Innovatio



SOLENT  
UNIVERSITY  
SOUTHAMPTON



Zachodniopomorski  
Uniwersytet  
Technologiczny  
w Szczecinie



With the support of the  
Erasmus+ Programme  
of the European Union



Master Thesis  
Master of Science  
Marine Technology  
Department of Hydrodynamics and Ocean Engineering

# Numerical and Parametric Optimization of Ship Hull Form

submitted on 16 August 2024

by

Smit DHANANI

**Supervisor :**

**Nolwenn Huon De Kermadec**



# TABLE OF CONTENTS

<b>Declaration</b>	<b>v</b>
<b>Abstract</b>	<b>vi</b>
<b>Acknowledgement</b>	<b>vii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Aim & Motivation . . . . .	1
1.2 Report Outline . . . . .	2
<b>2 Literature Review</b>	<b>4</b>
2.1 Review of CFD tools & Parametric models . . . . .	4
2.2 Review of Surrogate Model based Optimization . . . . .	6
<b>3 Optimization Tools &amp; Strategy</b>	<b>9</b>
3.1 Sampling Methods . . . . .	9
3.2 Surrogate Models . . . . .	12
3.3 Design Space Exploration & Exploitation . . . . .	16
<b>4 Parametric Models &amp; Optimization Framework</b>	<b>19</b>
4.1 Rhino based CAD . . . . .	20
4.1.1 Model Requirements & Objectives . . . . .	20
4.1.2 Free Form Deformation & Surface Transformation . . . . .	21
4.1.3 Python Script for Rhino . . . . .	25
4.1.4 Concept of Global & Local Optimization . . . . .	27
4.2 Fine Marine based Resistance Calculation . . . . .	28
4.2.1 Essential CFD Parameters . . . . .	28
4.2.2 Automation of CFD . . . . .	32
4.3 Python based Integration Framework . . . . .	33
4.4 Main user input file layout . . . . .	37
<b>5 Test Case I : Patrol Vessel</b>	<b>38</b>
5.1 Optimization Workflow & Setup . . . . .	38
5.1.1 Vessel Characteristics . . . . .	39
5.1.2 Objective function & constraints . . . . .	39
5.1.3 CFD setup . . . . .	39
5.2 Optimization of Fore Part . . . . .	42
5.2.1 Global cage modifications & Initial Sampling . . . . .	42
5.2.2 Results & comparison . . . . .	45
5.3 Optimization of Bulbous Bow . . . . .	47

## LIST OF FIGURES

---

5.3.1	Local cage modifications & Initial Sampling . . . . .	47
5.3.2	Results & Comparison . . . . .	50
<b>6</b>	<b>Test Case II : Cargo Vessel</b>	<b>53</b>
6.1	Optimization Workflow & Setup . . . . .	53
6.1.1	Vessel Characteristics . . . . .	53
6.1.2	Objective function & constraints . . . . .	54
6.1.3	CFD setup . . . . .	54
6.2	Optimization of Fore Part . . . . .	56
6.2.1	Global cage modifications & Initial Sampling . . . . .	56
6.2.2	Results & Comparison . . . . .	58
<b>7</b>	<b>Remarks &amp; Conclusion</b>	<b>62</b>
	<b>References</b>	<b>66</b>
	<b>Appendix A</b>	<b>67</b>
	<b>Appendix B</b>	<b>76</b>

## LIST OF FIGURES

1	Overall Optimization Workflow . . . . .	1
2	Comparison of Samping Criteria : ‘MAXIMIN’ (left) & ‘ESE’ (right) . . . . .	11
3	Comparison between RBF model(left) and Kriging model(right) . . . . .	13
4	K-fold Cross Validation Process . . . . .	15
5	Design Space Exploration(left) & Exploitation(right) . . . . .	17
6	Sphere within a cubic lattice Cage . . . . .	24
7	Cage Edit with Degree 1 . . . . .	24
8	Cage Edit with Degree 2 . . . . .	24
9	Workflow of Rhino Python Script . . . . .	25
10	Definition of Rhino Python Script . . . . .	26
11	Cage for Global Optimization (left) & Local Optimization (right) . . . . .	27
12	Mesh Refinement & Diffusion . . . . .	29
13	Workflow of CFD Automation . . . . .	32
14	Optimization Workflow . . . . .	33
15	Python Framework Architecture . . . . .	34
16	Surrogate Modeling and Bayesian Optimization workflow . . . . .	36
17	Patrol Vessel CFD Domain . . . . .	40
18	Global Design Parameters (Ba,Bb) for Patrol Vessel . . . . .	42
19	Design Space Sampling for Global Optimization of Patrol Vessel . . . . .	43

## LIST OF TABLES

---

20	Surrogate Model Comparison for Patrol Vessel . . . . .	44
21	Design Space Mapping and Contour for Global Optimization of Patrol Vessel .	45
22	Patrol Vessel Hull Lines (INITIAL & MODIFIED) . . . . .	46
23	Patrol Vessel Section Lines (INITIAL & MODIFIED) . . . . .	46
24	Patrol Vessel Global Wave Elevation Comparison . . . . .	47
25	Bulb Cage for Local Optimization of Patrol Vessel . . . . .	47
26	Illustrations of Bulb Modifications . . . . .	48
27	Design Space Sampling for Local Optimization of Patrol Vessel . . . . .	49
28	Surrogate Model Comparison for Patrol Vessel . . . . .	51
29	Bulbous Bow Section Lines (INITIAL & MODIFIED) . . . . .	52
30	Bulbous Bow Initial (Shaded) & Modified (Transparent) . . . . .	52
31	Cargo Vessel CFD Domain . . . . .	54
32	Global Design Parameters (Ba,Bb) for Cargo Vessel . . . . .	56
33	Design Space Sampling for Global Optimization of Cargo Vessel . . . . .	57
34	Surrogate Model Comparison for Cargo Vessel . . . . .	57
35	Design Space Mapping and Contour for Global Optimization of Cargo Vessel .	59
36	Cargo Vessel Hull Lines (INITIAL & MODIFIED) . . . . .	60
37	Cargo Vessel Section Lines (INITIAL & MODIFIED) . . . . .	60
38	Cargo Vessel Global Wave Elevation Comparison . . . . .	61

## LIST OF TABLES

1	Parameters for the Latin Hypercube Sampling Plan . . . . .	10
2	Criteria for the Distribution of Sample Points . . . . .	11
3	Local & Global Cage Edit Parameters . . . . .	23
4	CFD Parameters . . . . .	31
5	config.ini - input file layout . . . . .	37
6	Characteristics of a Generic Patrol Vessel . . . . .	39
7	Optimization Table for Minimizing Surge Resistance . . . . .	39
8	Patrol Vessel CFD Parameters . . . . .	41
9	Global Cage Edit Parameters for Patrol Vessel . . . . .	43
10	Global Optimization results for Patrol Vessel . . . . .	46
11	Local Cage Edit Parameters for Patrol Vessel . . . . .	49
12	Local Optimization results for Patrol Vessel . . . . .	51
13	Characteristics of Cargo Vessel . . . . .	53
14	Optimization objectives & constraints . . . . .	54
15	Cargo Vessel CFD Parameters . . . . .	55
16	Global Cage Edit Parameters for Cargo Vessel . . . . .	56
17	Initial Resistance Predictions and Deviations . . . . .	58

LIST OF TABLES

---

18 Global Optimization results for Cargo Vessel . . . . . 59

## DECLARATION OF AUTHORSHIP

I declare that this thesis and the work presented in it are my own and have been generated by me as the result of my own original research.

Where I have consulted the published work of others, this is always clearly attributed.

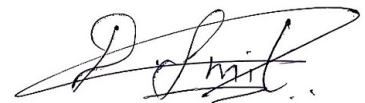
Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.

I have acknowledged all main sources of help.

Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

This thesis contains no material that has been submitted previously, in whole or in part, for the award of any other academic degree or diploma.

I cede copyright of the thesis in favour of the University of Liege and École Centrale de Nantes.



16 August 2024

Smit DHANANI

## ABSTRACT

Optimization has become a crucial part of modern design practices, with advancements in numerical methods enabling designers to analyze multiple designs and identify the best options. However, the iterative nature of optimization can be time-consuming and computationally demanding, especially for complex designs with numerous criteria and constraints. This challenge is particularly evident in hull form optimization, where design possibilities are limitless.

To address this issue, a surrogate-based optimization strategy is proposed. This approach can replace expensive numerical simulations with an equivalent surrogate (or meta) model, trained using data from a limited set of initial simulations. The model can be further refined using suitable optimization algorithms.

Implementation of this strategy however, requires seamless interaction among various tools integrated within a common framework. This thesis presents the development of such a framework for hull form optimization. It integrates parametric designs, numerical methods, surrogate models, and optimization algorithms under a common Python-based framework to streamline the optimization process.

**Keywords:** Hull form optimization, Surrogate based optimization, Bayesian optimization, Parametric models, Surrogate models, Free form deformation, Design space.



## **ACKNOWLEDGEMENT**

I would like to express my profound gratitude to my supervisor Nolwenn Huon De Kermadec for her kind faith, relentless support and unwavering encouragement throughout the course of this thesis. I am also grateful to Mr. Vincent Seugin for providing me the opportunity to pursue my thesis at MAURIC, and to all my colleagues there for their expert consultation and support. My heartfelt thanks go to Prof. Nahon for introducing this topic and helping me truly gauge its merits and implications.

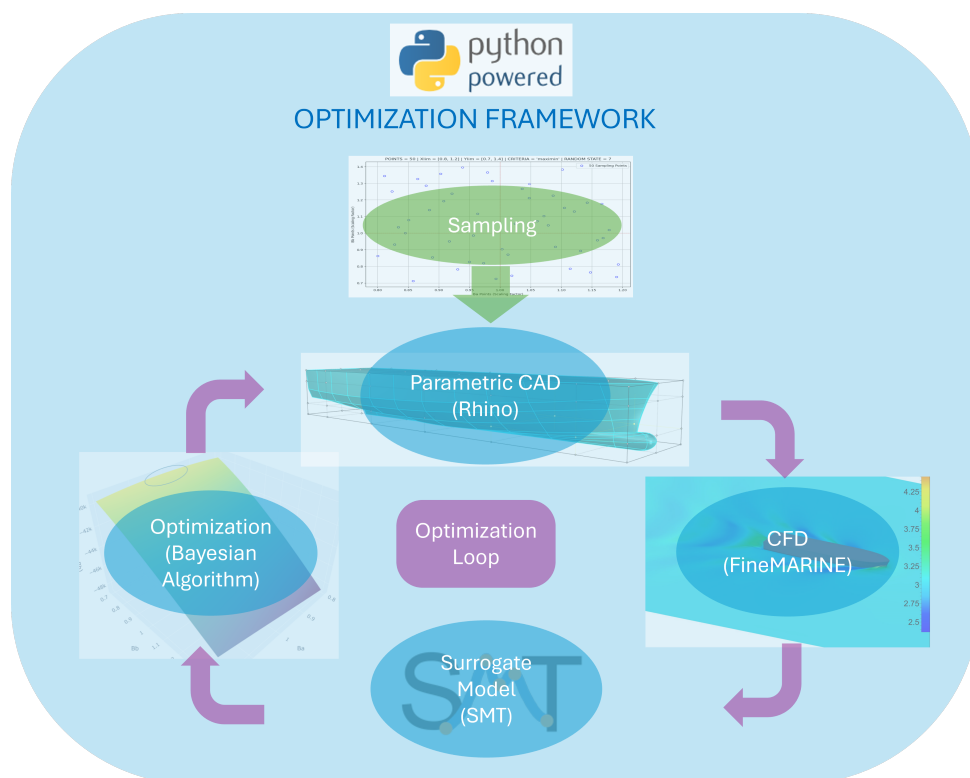
In addition, I would also like to thank Prof. Rigo for whom I share my greatest respect and appreciation. His tireless efforts in shaping the EMSHIP+ program and fostering the success of its students are truly remarkable and inspiring.

Finally, a special vote of thanks goes to my friend Abhemanyu for sharing his valuable feedback and critical remarks, which significantly improved the articulation and coherence of this report.

# 1 INTRODUCTION

## 1.1 Aim & Motivation

There can be multiple ways to approach and thus solve an optimization problem based on the problem definition and expenditure of resources to perform optimization. However, the basic principle remains consistent in most cases which is largely governed by some iterative process guided by an optimization algorithm. Within this process, each subsequent iteration seeks to find a design better than the previous iteration and the process continues until a set of user defined criteria are met or some convergence is achieved. This description broadly encapsulates the overall strategy of an optimization workflow. It may also be possible as is the case in most of the problems, that a single loop of iteration may depend on series of processes and their sub-processes each of which may in-turn be an optimization problem in itself. This is especially true in the present work where multiple tools of independent nature are a part of the design chain. Moreover, for these tools to be able to efficiently interact with each other, they must be seamlessly integrated together under a common framework.



**Figure 1:** Overall Optimization Workflow

Thus, the work carried out as a part of this thesis is an attempt to develop a common framework using ‘Python’ as the programming language. The choice of this particular language was attributed to its open-source nature and its versatility with other tools and softwares which are a part of the larger framework.

Figure 1 provides a vivid description of the optimization framework and the workflow of various processes which are a part of it. The architecture of this framework ensures homogenous integration of four essential components which together form an optimization loop. These four components are :

- Parametric design models - developed using Rhinoceros3D software.
- Computational Fluid Dynamics (CFD) analysis - performed using FineMARINE software.
- Surrogate/Meta models - generated using Surrogate Modeling Toolbox (SMT)
- Optimization models - created using Bayesian Optimization Toolbox.

More details about the use, implementation and integration of these models will be provided in the subsequent chapters.

### 1.2 Report Outline

The report was divided into seven chapters where the first four chapters describe the development of the framework, followed by next two chapters on the implementation and the final chapter concludes the report.

1. The current chapter lists various aspects of the contents within the report as well as gives the reader an overall idea of the general strategy adopted herein for hull form optimization.
2. Second chapter presents a comprehensive review of research in the domain of optimization. It highlights the development of parametric hull models and the use of numerical methods to derive better hull forms. Moreover, the chapter also elaborates on some of the recent advancements in surrogate based design and optimization methods and justifies their applicability within the current framework.

3. Third chapter aims at familiarizing the reader with some fundamental aspects related to development of surrogates or meta models and draws some general guidelines on the use of optimization algorithms within the current context. While the scope of this domain remains quite vast, an attempt was made to present relevant topics in a fairly condensed manner.
4. Fourth chapter gives a vivid description of a python based integration framework which couples parametric models and surrogate models with popular optimization algorithms. Moreover, majority of the work as a part of the thesis was dedicated to the development and testing of this framework. Thus, this chapter attempts to explain the architecture and working principle of the developed framework, focusing on its novel aspects. The author excludes detailed discussions of general CFD or CAD practices that are presumed to be familiar to the reader.
5. Fifth chapter presents a first test case study on optimization of a hull form based on a generic patrol vessel. It gives an opportunity to present the optimization workflow in an orderly manner and also serves as reference guide for users of the Python-based framework. The reader is encouraged to thoroughly follow this chapter to gain better insight into the concepts as described in the preceding chapters. Appendix A & B further compliments additional details pertaining to these two chapters.
6. Sixth chapter follows a similar study as the previous chapter, but with a different hull form design based on a cargo vessel. The objective of this study was to test the robustness of the framework for a different type of hull form and to identify any potential bugs which may have otherwise gone unnoticed in the previous case. Moreover, it presents the reader with some interesting results and shortcomings of the optimization process.
7. Seventh chapter concludes the report by postulating some remarks on the advantages & limitations of the framework. It also delivers suggestions pertaining to future scope and ideas for potential enhancements of the developed framework.

## 2 LITERATURE REVIEW

Optimization of hull forms has been a subject of persistent research since the early days of the modern shipbuilding. Much of the research during those early days was driven by the need to improve vessel stability, seakeeping characteristics, and most importantly, vessel speed to shorten voyage times. This led to the evolution of slender ship designs with elongated hulls and relatively shorter beams. These designs were based on traditional rudimentary practices, trial and error methods, and extensive testing of hull models in towing tanks. To further our understanding, this sections presents a comprehensive review of parametric models, numerical codes, surrogate modeling and optimization algorithms within the context of improved hull form design.

### 2.1 Review of CFD tools & Parametric models

Advancements in fluid dynamics led to the development of mathematical theories and models to better understand the hydrodynamic aspects of ship design. A key breakthrough in this evolution was the introduction of potential flow theory and the Reynolds decomposition of Navier-Stokes equations to model complex flow behavior around vessels. This furthered our understanding of various components of total resistance, which were incorporated into the design process. Additionally, tremendous growth in computational power led to the development of computational fluid dynamics (CFD), enabling designers to model fluid flow around a vessel and estimate the forces acting on its surface.

In recent decades, the objective of optimization has revolved around improving the fuel and operational efficiency of vessels. Modern design strategies involve using multiple computational tools coupled with enhanced optimization algorithms to evaluate an entire spectrum of design possibilities and narrow down promising designs. In this context, a review of various optimization practices suggested by academic researchers and followed by industrial designers alike was conducted and thoroughly analyzed.

Majority of the studies carried out in this field relied on the use of commercial softwares for CFD calculations. Moreover, the due to the computationally

demanding nature of these calculations, most of researchers relied on the use of less expensive potential flow theory or boundary element method to estimate the surge resistance while some implemented a Reynolds averages Navier-Stokes equation (RANSE) based model with a coarse mesh quality for their analysis. However, this approach has been found unreliable because the errors in the estimated numerical results often exceed the actual differences in resistance values between consecutive designs. Feng *et al.* [1] proposed a fully parametric modeling technique and enumerated various design parameters for optimization of three containership designs: Duisburg test case (DTC), Kriso containership (KCS) and S-175 container vessel. Similarly, Han *et al.* [2] in their study validated the effectiveness of hydrodynamic optimization on parametric hull forms. In both cases however, the authors relied on the use of a potential flow solver provided by a commercial software ‘SHIPFLOW’ to validate their results. This approach as mentioned earlier is not reliable due to the inability of the underlying numerical code to accurately capture the flow physics.

The ‘Twenty-First Symposium on Naval Hydrodynamics - 1997’ [3] in one of their chapters published an extensive study on the use of computational numerical tools for optimization of a Series 60 hull form, and co-related their results with experimental values. Although they obtained an optimized design with a CFD based resistance deduction of 11 to 9%, the experimental results of this optimized design in-fact found an increase in the resistance contradicting their CFD results. The authors attributed this contradiction to the limitation of the CFD solvers and their inability to capture the viscous effects on stern wave making behavior. Furthermore, to reduce the computational load, they used a zonal approach method which divides the fluid domain into three regions, which are solved individually using potential flow, boundary layer and RANSE methods. This was however an early study on the use of computational tools to iteratively modify a hull form until an optimum is reached. Significant advancements in numerical codes have taken place since then making them more accurate and reliable.

Similarly, many studies have been carried out to investigate the effects of localized surfaces such as bilge keels, bulbous bow, skeg, etc. on the overall flow behavior around the vessel. Čerka *et al.* [4] analyzed multiple skeg designs

for optimization of a catamaran-type research vessel and suggested rounded skeg edges with sharp ends to ensure smooth flow separation from the skeg and transition to the propeller.

Likewise, Kracht [5] analyzed the sensitivity of multiple bulbous bow parameters with respect to various resistance components using the linearized theory of wave resistance and some experimental results. They found the volume, sectional area at the fore perpendicular, and the protruding length of the bulb to be the most important quantities governing the resulting effect on the bow wave system. The study also concluded that the amplitude of the bow wave is a function of the volume of the bulb and laid out general guidelines for dimensioning of a bulbous bow. Moreover, a bulbous bow is designed for a specific type of vessel and under a well-defined set of operating conditions within which it delivers its intended performance. Any significant change in the operational profile of the vessel could inhibit or may even degrade the overall performance of the vessel.

### **2.2 Review of Surrogate Model based Optimization**

To address this issue of designing a hull form that can deliver optimum performance across a multitude of operating conditions, researchers dwelled into the use of multi-objective optimization algorithms wherein the design objective was to minimize/maximize the desired characteristics (such as resistance, seakeeping, etc.) for multiple conditions (such as multiple speeds, sea-states, etc.). However, it must be noted that there is an exponential rise in the discretization of the design space as the number of variables and objectives increases. Thus, it becomes infeasible even with modern computational tools to identify the merit of each design individually. This leads us to the use of surrogate models where an input-output relationship between objective functions and design variables is established using a small set of design samples. The primary use cases of such models could be traced back to automotive and aerospace industries which deal with a large set of often contradictory variables to find the best possible design. In recent decades, however, thanks to the advancements in meta-modeling techniques and neural networks, surrogate model-based design and optimization techniques have been widely adopted by many industries.

The subsequent discussions will elaborate on the applicability of surrogate models in hull form optimization. Furthermore, some keywords might find repeated usage and the author would like to explicitly clarify that, while the term multi-fidelity can convey different classifications of numerical simulations, a widely accepted notion is that ‘low-fidelity’ refers to computationally less demanding calculations, whereas ‘high-fidelity’ may refer to more computationally intensive calculations. In this context, a ‘potential flow’ based solver can be regarded as low-fidelity while a ‘viscous flow’ based solver can be associated with high-fidelity. On similar grounds, even a RANSE simulation can be identified as low or high fidelity based on the quality and refinement of its mesh.

Liu *et al.* [6] investigated the use of surrogate models to predict the resistance and thus optimize the hull form. They validated the use of a multi-fidelity Co-Kirging model which was developed using a large set of initial samples from a low-fidelity potential flow theory-based solver and a smaller sampling set of high-fidelity viscous flow theory simulation. They concluded that the multi-fidelity approach resulted in lesser computational time and better quality of the surrogate models by leveraging the efficiency of potential flow theory and the accuracy of viscous flow theory.

Wang *et al.* [7] evaluated the Gaussian process regression algorithm coupled with an adaptive sampling strategy to perform a surrogate-based design optimization of a deep-sea aquaculture vessel. A key feature of this was the use of sequential sampling, specifically in the region of interest so as to efficiently utilize every single design point during optimization. Their study found that the adaptive sampling methodology resulted in better uncertainty quantification of the surrogate model and 46.67% improvement in optimization efficiency over conventional surrogate-based design practice. This could further be co-related with the concept of design space exploration and exploitation proposed by the Bayesian optimization algorithm.

Jakub and Radomil [8] performed an extensive review of surrogate models for their suitability for FEM and CFD-based problems to various industrial use cases. Their recommendations were widely adopted throughout this report and the reader might find frequent mentions of this paper in subsequent chapters. Likewise, the



theoretical explanation is given by Forrester *et al.* [9] in their book *‘Engineering design via surrogate modeling: a practical guide’* and Jiang *et al.* [10] in *‘Springer Tracts in Mechanical Engineering Surrogate Model-Based Engineering Design and Optimization’* have provided detailed scientific information covering all the aspects of surrogate models and their wider implementation.

Many commercial software for scientific computing such as ‘MATLAB’ offer surrogate modeling features. In this regard, one of the most widely adopted tools used and recommended by researchers is a Python-based open source tool: *‘Surrogate Modeling Toolbox’* by Saves *et al.* [11] which offers a wide spectrum of models to choose from along with many sampling techniques and optimization algorithms. Likewise, the *Bayesian Optimization Toolbox* developed by Nogueira [12] offers a host of optimization techniques based on the Gaussian process regression method and is been widely implemented in academia for all types of optimization problems. These toolboxes have been implemented in the present work and the reader is advised to refer to their documentation for further references.

The information synthesized by this literature review enabled the author to draw a clear roadmap of the tasks to be performed throughout the thesis. In addition, the familiarization of key ideas and relevant topics facilitated the conceptualization of the overall optimization framework which was developed as a part of the thesis.

### 3 OPTIMIZATION TOOLS & STRATEGY

This chapter gives a description and implementation of various open source tools and modules used in the current optimization framework.

#### 3.1 Sampling Methods

The primary use of sampling methods could be traced back to the formulation of Design of Experiments (DoE) in which the design iterations were capped by the limitation on the number of experiments that could be performed to test and validate each new design. Moreover, the larger issue of the ‘Curse of Dimensionality’ which refers to an exponential increase, in the number of sampling points as the number of design variable increases makes it impossible to perform a full factorial DoE. Furthermore, the accuracy of the surrogate model built thereafter would also depend on the distribution and some specific criteria related to the sampling points. Thus, in order to efficiently utilize the budgeted outlay of experiments, the sampling method must be able to generate a set of points which:

- Covers the entire permissible design space
- Evenly distributes the sampling points in a multidimensional design space
- Satisfies the user defined constraints and limitations of each design parameter

The sampling methods could further be classified as *one-shot sampling* and *adaptive sampling*. As the name suggests, one-shot sampling is used to generate all the points at the same time such that the space filling criteria is met. This kind of sampling is used to approximately map out the design space and in cases where the initial setup largely remains unchanged for the subsequent experiments. Adaptive sampling on the other hand iteratively generates new sample points based on the information and results available from the previous points.

The sampling methodology adopted in the present thesis generates an initial set of points to explore and thus map out the design space after which a preliminary surrogate model is created using these initial points. This preliminary model may give a rough idea of the input-output relationship of the surrogate model.

The model may need to be further reinforced by iteratively adding new points in a stochastic manner as a function of the data generated from previous points. This transition of an optimization workflow from exploring the entire design domain to exploiting a specific region of potential maxima or minima is the most important governing factor in the entire process.

One of the most widely used sampling method is the *Latin Hypercube Sampling (LHS)* which results in better space filling characteristics in a generalized higher dimensional design space called ‘Latin Hypercube’ Helton and Davis [13]. An open-source Python based package *Surrogate Modelling Toolbox (SMT)* was used to generate the sampling points and build the surrogate model. This tool is widely used across multiple domains concerning surrogate modelling and is tested with several benchmarking problems. The reader is encouraged to refer to detailed documentation of *SMT* available at [smt.readthedocs.io](http://smt.readthedocs.io).

The Latin Hypercube Sampling module provided by SMT primarily takes three inputs from the user as described in Table 1 to generate a set of sampling points.

**Table 1:** Parameters for the Latin Hypercube Sampling Plan

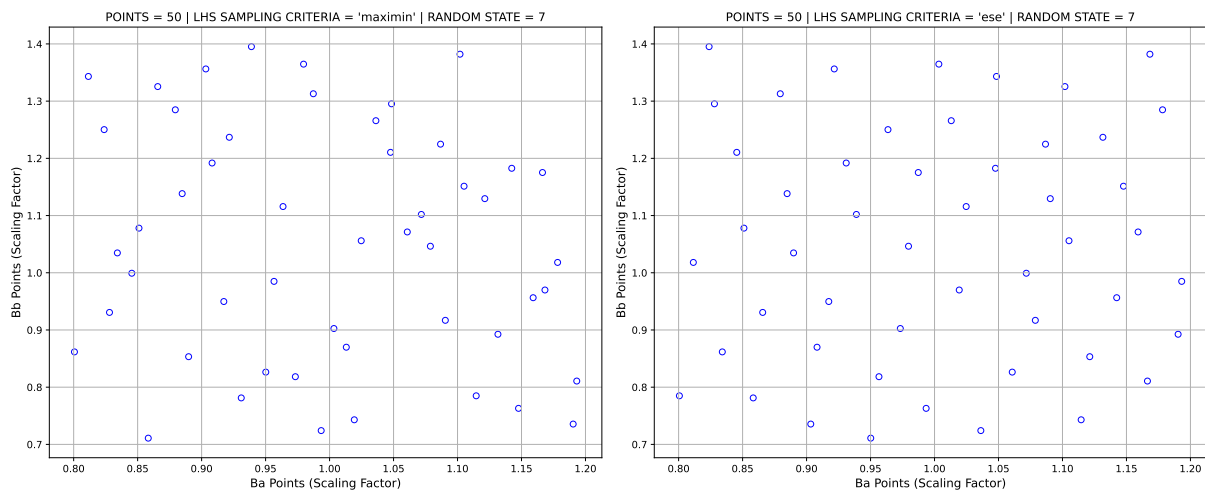
Parameter	Description
<code>xlimits</code>	An array of size $(n \times 2)$ defining the upper and lower bounds of $n$ design variables.
<code>criterion</code>	The governing criteria for the distribution of sample points.
<code>random_state</code>	A seeder for future reproducibility of the sampling plan (since LHS draws a unique set of points each time the function is executed).

As discussed earlier, the efficiency of the optimization workflow is highly sensitive to the initial sampling points which in-turn is governed by the LHS ‘criterion’ given by the user. SMT offers a list of 5 comprehensive criteria for the user to choose from and each criterion is based on a well defined set of sampling distribution. A brief summary of these criteria can be found in Table 2.

**Table 2:** Criteria for the Distribution of Sample Points

Criterion	Description
center	Center the points within the sampling intervals.
maximin	Maximize the minimum distance between points and place the point in a randomized location within its interval.
centermaximin	Maximize the minimum distance between points and center the point within its interval.
correlation	Minimize the maximum correlation coefficient.
ese	Optimize the design using the Enhanced Stochastic Evolutionary algorithm (ESE).

Before selecting the criterion, it is important to get an understanding of the spatial distribution of points as a function of these criteria. These points can be produced for multiple parameters resulting in a multi-dimensional design space. However, for the sake of simplicity and visual understanding of the distribution, a comparison between two criteria : ‘maximin’ & ‘ese’ for a 2D sampling plan is made in Figure [2]. It can be observed that the ‘ese’ criteria results in better distribution & space filling characteristics compared to ‘maximin’ criteria. Likewise, a similar comparison was made between the rest of criteria and ‘ese’ was found to be better than most of them. Moreover, the choice of ‘ese’ was coherent with the recommendations given by Jin *et al.* [14] wherein they found ‘ese’ to outperform other algorithms in terms of convergence & computational efficiency.

**Figure 2:** Comparison of Samping Criteria : ‘MAXIMIN’ (left) & ‘ESE’ (right)

#### 3.2 Surrogate Models

A broader connotation of the term ‘Surrogate models’ or ‘Metamodels’ essentially relates to the development of mathematical tools used to establish some relationship between the set of input data and its corresponding output data. This can be achieved by many ways of varying complexity from simple analytical equations to sophisticated deep neural networks. The primary objective of these models is to eventually be able to replicate the behaviour of the original phenomenon on which it is trained or derived. This replicative nature of surrogate models is of significant importance in engineering discipline especially for optimization problems which are heavily dependent on expensive processes such as physical experiments or high-fidelity computational simulations.

In the context of current problem - ‘Hull form Optimization’, surrogate models can be efficiently employed to emulate the computationally expensive CFD simulation of each hull form design. While surrogate models cannot exactly replicate the results produced by the simulations, they can still reveal useful information about the general behavior of the concerned design domain. Once a surrogate model with sufficient accuracy is trained and validated, it can be used to

- Perform sensitivity analysis of the design with respect to desired input parameters
- Extract gradient related information of the design space
- Serve as an objective function for other optimization algorithms

Surrogate models have been widely discussed in the academic literature and has continued to remain an area of active research due to their ability to efficiently approximate complex systems. While numerous models have been developed so far, each with its own advantages and limitations, they are typically tailored to a specific type of problem by inculcating the underlying physics. Although exploring the full spectrum of available models is beyond the scope of this thesis, some models show promising results and are particularly suitable for engineering applications. In this regard, Radial Basis Functions (RBF) and

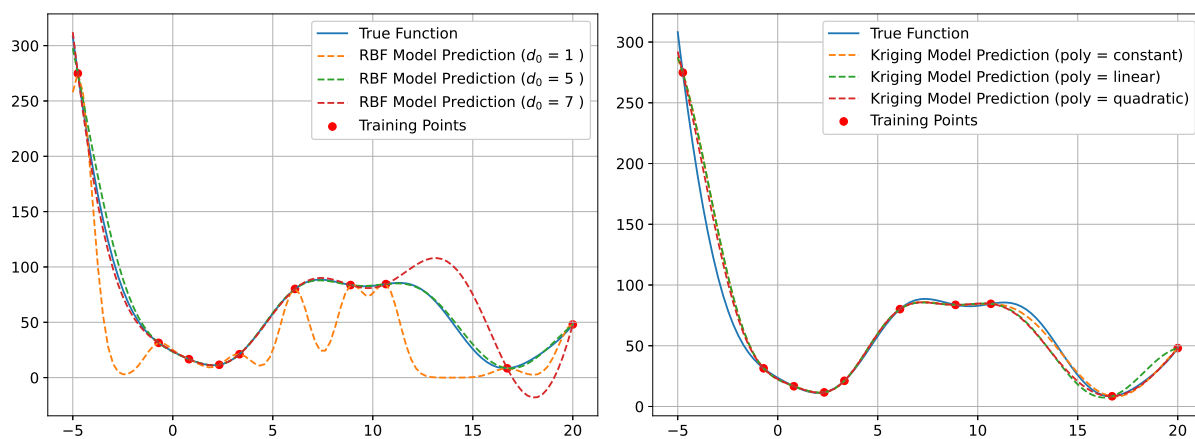
Kriging models have seen wider adaptability in CFD and Finite Element Method (FEM) based optimization problems as pointed out by Jakub and Radomil [8]. The Python-based *Surrogate Modelling Toolbox (SMT)* offers a range of surrogate models to choose from with a particular focus on estimating derivatives for gradient based optimization.

### RBF and Kriging Model

This section draws a brief comparison of RBF and Kriging model with respect to a common benchmarking problem (*Branin function*) defined by equation 1. Both of these models were built against the same set of training points. Moreover, each model has its own set of input parameters which can significantly alter their behaviour.

$$f(\mathbf{x}) = \left(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos(x_1) + 10 \quad (1)$$

where  $\mathbf{x} = (x_1, x_2)$  with  $-5 < x_1 < 20$ ,  $0 < x_2 < 20$



**Figure 3:** Comparison between RBF model(left) and Kriging model(right)

- The functions defining both RBF and Kriging model are shown by equation [2] & [3] respectively. These models are a function of multiple dependent variables, each of which determines the resulting nature of the model and the reader is encouraged to refer to their detailed description available at [smt.readthedocs.io](http://smt.readthedocs.io).

- For the sake of simplicity, the RBF model was evaluated with respect to its scaling parameter  $d_0 = [1 \ 5 \ 7]$  while the Kriging model was studied by varying its deterministic term  $\sum_{i=1}^k \beta_i f_i(x)$  which can have a constant, linear or quadratic nature.
- The rest of the variables were set to their default values as suggested by the *SMT toolbox*.
- Fig [1] shows a comparison of both of these models and it can be observed that RBF model is more sensitive than Krigin model. This can be interpreted in terms of adaptability of the model with respect to the problem that they try to emulate. In that sense, RBF model is more suitable due to its ability to capture the irregular and unexpected trends within the function. Likewise, Kirging model is suitable in estimating the general trend of a function. In practice however, the true function is never known and thus each of these models can be utilized simultaneously by training them with the same dataset and estimating their relative divergence.
- This method of utilizing several surrogate models and combining them through a weighted sum is referred to as ‘Ensembles of Surrogate Models’. This ensures that the final results are not biased by any single model and the true function is well captured both at global and local level.

$$\text{RBF Model: } y = p(x)w_p + \sum \phi(x, x_{t_i})w_r \quad (2)$$

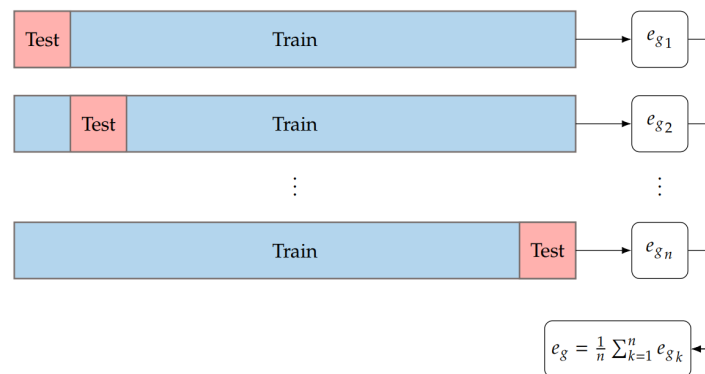
$$\phi(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{d_0}\right)$$

$$\text{Kriging Model: } y = \sum_{i=1}^k \beta_i f_i(x) + Z(x) \quad (3)$$

### Validation of Surrogate Model

Once a model is built, it must be validated against a set of already known data-points. The most common methodology to validate the accuracy and

reliability of a model is by estimating the divergence of model-predicted values with the true values. This requires splitting the initial dataset into training data and testing data whereby the model is first developed using the training data and it is then validated using the testing data. In the validation phase, the model is fed with the same input points of the testing dataset and its prediction (model output) is compared with the output values of the testing dataset. While there are no defined methods to allocate the relative proportion of train-test data-split, but a widely adopted rule of thumb is to split the training and testing data into a 70-30 proportion respectively. This method of cross validation works well with the abundance of available dataset. However, when it is difficult or expensive to generate the necessary data, as is the case for high-fidelity simulations, it becomes important to efficiently utilize every single data-point without significantly compromising on validity of the model. Moreover, the need for a larger dataset in-turn contradicts the use of surrogate models in the first place. For this reason, a *k-fold cross validation* methodology was adopted in the present work of Hull form Optimization. The general procedure of this methodology was adopted from the chapter ‘Surrogate Based Optimization’ in a book by Martins and Ning [15] and is highlighted in figure 4



**Figure 4:** K-fold Cross Validation Process

1. Randomly split the initial data in  $n$  sets
2. Utilize  $n-1$  data-points to train the model and the remaining  $1$  point to cross validate.
3. Repeat this process for  $n$  configurations such that all possible  $n$  validation points gets utilized.



4. Estimate the averaged generalized error of the model corresponding to each of the  $n$  configurations.
5. Finally, choose a model configuration with the least averaged generalized error.

#### 3.3 Design Space Exploration & Exploitation

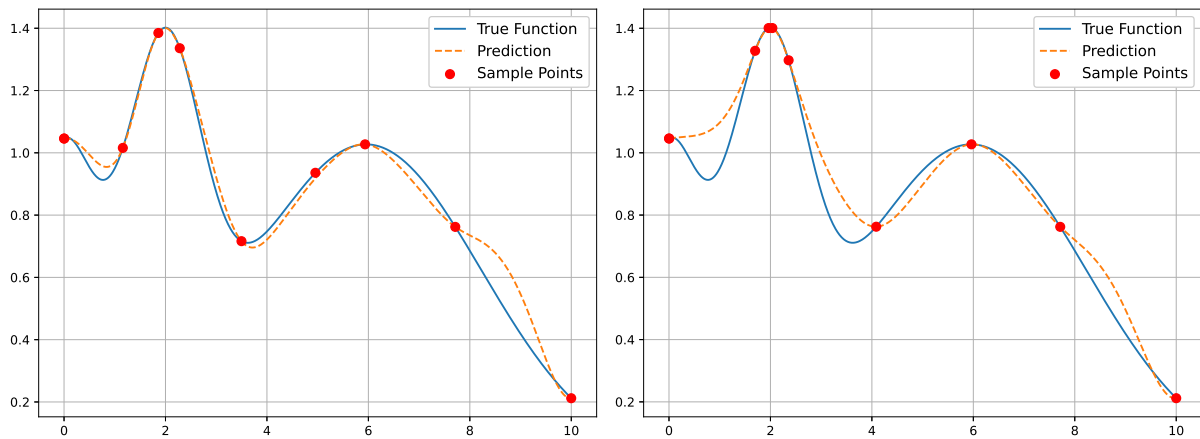
Now that the basic premises of the development and use of surrogate model is established, we can dwell deeper into the use of Bayesian optimization algorithm. A Python-based open-source package developed by Nogueira [12] was used as an external toolbox for implementation of Bayesian optimization algorithm within the current hull form optimization framework. Its detailed documentation can be found at [bayesian-optimization.github.io](https://bayesian-optimization.github.io)

Bayesian optimization algorithms are employed to reinforce preliminary surrogate models by iteratively refining their accuracy and predictive capabilities. This process involves the use of infill criteria, which guide the selection of new sampling points to improve the model's performance. A stochastic or adaptive sampling approach was utilized, allowing the model to dynamically adjust its sampling strategy based on the evolving understanding of the design space. Initial design space exploration was conducted using a preliminary sampling plan as highlighted in section 3.1, which provides a broad overview of the function's behavior. Subsequently, the iterative Bayesian optimization strategy focuses on exploitation, systematically refining the surrogate model by targeting areas of interest identified during the exploration phase. This combined exploration and exploitation approach ensures a comprehensive and efficient optimization process, ultimately leading to more accurate and robust surrogate models.

The selection of an appropriate acquisition function is crucial in balancing exploration and exploitation within Bayesian optimization. The following acquisition functions are provided by the optimization toolbox.

- Upper Confidence Bound
- Expected Improvement
- Probability of Improvement

Amongst these, the Expected Improvement function was chosen for its compatibility with Kriging and RBF models. The behavior of this function is governed by the parameter ( $\xi$ ), which influences the trade-off between exploration and exploitation. A value of  $\xi = 0.1$  promotes exploration, while  $\xi = 0.0$  favors exploitation. In this study,  $\xi$  was set to 0.0 during the adaptive sampling process to prioritize exploitation. Furthermore, the model's accuracy is directly correlated with the number of sampling points allocated for both the exploration and exploitation phases. This allocation is determined by the limitation of the computational budget i.e. total number of sampling points available for the optimization problem. This trade-off is essential to achieve an optimal balance between model accuracy and computational efficiency.



**Figure 5:** Design Space Exploration(left) & Exploitation(right)

- Figure 5 illustrates the comparative analysis of Design Space Exploration ( $\xi = 0.1$ ) versus Exploitation ( $\xi = 0.0$ ) utilizing the ‘Expected Improvement’ acquisition function.
- The true function, defined by Equation 4, served as the basis for this analysis which is plotted on the vertical axis of Figure 5. The Bayesian Optimizer was allocated 9 iterative sampling points within the domain  $[0, 10]$  which are plotted on the horizontal axis of Figure 5.

$$f(x) = e^{-(x-2)^2} + e^{-\frac{(x-6)^2}{10}} + \frac{1}{x^2 + 1} \quad (4)$$

- Observational analysis reveals a distinct clustering of points near local

maxima during the exploitation phase, contrasting with a more uniform distribution across the domain during exploration.

- Optimal performance can be achieved through judicious allocation of sampling points between exploration and exploitation phases, thereby striking an effective balance between broad search and focused refinement.

### 4 PARAMETRIC MODELS & OPTIMIZATION FRAMEWORK

This chapter explores the integration of Parametric Computer-Aided Design (CAD) and Computational Fluid Dynamics (CFD) model within a unified framework.

A key innovation in this approach is the creation of a ‘Python Based Integration Framework’. This framework leverages the Python scripting capabilities inherent in both Rhino (CAD software) and FineMARINE (CFD software), thereby serving as a bridge between the CAD and CFD environments. The integration allows for automation and seamless synchronization of various processes and sub processes involved in each iteration of the design cycle, enabling a more efficient and cohesive workflow between geometric modeling in Rhino and resistance estimation in FineMARINE.

The Python-based framework facilitates several critical functions which are in the following order:

1. Modifying the hull geometry in the CAD model based on optimization parameters
2. Seamless transfer of updated geometry from Rhino to FineMARINE
3. Automated setup and execution of CFD simulations in FineMARINE
4. Extraction and processing of simulation results for creation of a Surrogate Model
5. Refining the surrogate model by adaptive sampling based on Bayesian Optimization
6. Finally, replacing the computationally expensive CFD simulations by the surrogate model and evaluating the optimum point of this model.

Furthermore, the chapter discusses the coupling of this integrated CAD-CFD system with a Bayesian Optimization Toolbox. This coupling introduces an intelligent optimization layer to the framework, enabling efficient exploration of design spaces and optimization of performance parameters, particularly in the context of surge resistance analysis.

This integrated approach is derived from practices in the aerospace and automotive industries, and similar trends are now being observed across the naval industry, combining parametric modeling, high-fidelity simulations, and optimization techniques within a single unified framework.

### 4.1 Rhino based CAD

Rhinoceros 3D (Rhino) is a CAD application with advanced surface modelling capabilities based on NURBS (Non-uniform rational B-spline) mathematical model to represent curves and surfaces. This enables designers to model complex and free-form geometries with a high degree of accuracy. Furthermore, the ‘cage edit’ functionality offered by Rhino allows users to perform smooth surface transformations and refinement. This can be coupled with python scripting feature of Rhino to rapidly perform precise surface modifications over multiple iterations of a design. Thus, due to its versatility, compatibility and wider adaptability, Rhino was chosen as the primary CAD engine to develop a partially parametric model of the ship hull form.

#### 4.1.1 Model Requirements & Objectives

A common practice followed by most design offices is to derive new hull lines from existing parent vessels or sister ships by modifying them to meet specific owner requirements and operational profiles. While fully parametric models offer comprehensive control over hull design, their development is time consuming and highly demanding. Moreover, these models often lack flexibility and are typically tailored to specific ship types.

In contrast, a partially parametric model present a more efficient alternative, particularly when new designs deviate only slightly from an existing parent design. These models offer greater flexibility, require less development time and are especially useful for targeted modifications to specific areas of the hull, such as the bulbous bow, skeg region, transom area, or the fore/aft part of the vessel. In context of the present work, the following requirements were laid out for the parametric model:

- The model must be suitable and adaptable to all types of mono-hulls.

- Only the existing surfaces or patches within the existing CAD file (.iges or .3dm) must be modified and no new surfaces must be added or removed.
- The manufacturability of the resulting hull form must be respected, avoiding any *double curvature surfaces*.
- The model must take into account externally imposed additional constraints, such as minimum clearance for *bow thruster tunnels*, *gearboxes*, or any additional user-specified objects.
- The resulting hull form must form a *closed polysurface* without any *naked edges* or *non-manifold edges*.
- The transformed surfaces must be smoothly integrated with the neighboring surfaces.
- The model must be able to accurately calculate the hydrostatic properties of the vessel.

Considering these factors, the decision was made to develop a framework suitable for creating a partially parametric CAD model that can readily adapt to various input mono-hull designs. This model is based on the principle of free-form deformation and was implemented using the cage edit feature in Rhino. By adopting this methodology, designers can efficiently modify existing hull forms while maintaining the ability to make precise, localized adjustments.

### 4.1.2 Free Form Deformation & Surface Transformation

Free Form Deformation (FFD) is a technique which was originally proposed by Sederberg and Parry [16] to perform deformation of solid geometric models. It has been refined over the years and has found significant relevance for parametric shape optimization problems. The basic principles governing this technique were adopted from Reid [17] and Samareh [18].

- A three-dimensional grid (FFD box) is constructed, with its vertices serving as volumetric control points.

- The target geometry, such as a ship hull or a specific region like the bulbous bow, is encapsulated within this control grid.
- A mapping function  $\mathbb{R}(u, v, w)^3 \rightarrow \mathbb{R}(x, y, z)^3$  is defined, establishing a relationship between the spatial coordinates of the encapsulated surfaces  $(x, y, z)$  and the control points of the grid  $(u, v, w)$ .
- Finally, Surface deformations are achieved by manipulating the control points  $(u, v, w)$  using appropriate magnitude and direction vectors.

The efficacy of FFD lies in its ability to parameterize the control points rather than the geometry itself, allowing for efficient and flexible surface transformations. The smoothness and continuity of these perturbations are governed by the chosen mapping function and the applied deformation vectors. This FFD technique can be implemented using the *Cage Edit* transformation feature in Rhino.

This *Cage Edit* feature has numerous user-input parameters that allow for precise control over the deformation process. These parameters include the type of cage (Linear, Planar, Cubical, Bounding Box, or user-defined), deformation accuracy, number of cage points along each axis, and the degree of modification in each direction. For a comprehensive understanding of these options, readers are encouraged to refer to its detailed documentation available at [*Cage Edit Documentation*]. Furthermore, to optimize the use of this feature for hull form modifications, a preliminary study was conducted to determine the most suitable set of parameters for both global and local level adjustments.

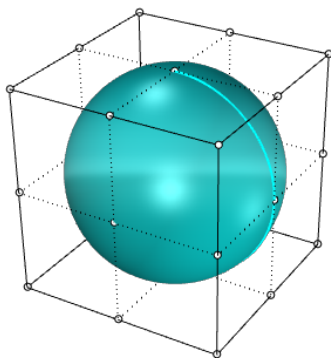
This systematic approach resulted in the identification of specific parameter combinations that produced the desired surface characteristics and are summarized in Table 3. It must be noted that the term ‘Degree of Points’ refer to the degree of Non-uniform rational B-spline (NURBS) polynomial that forms the edges of the cage. Its implications are illustrated in Figure 6.

**Table 3:** Local & Global Cage Edit Parameters

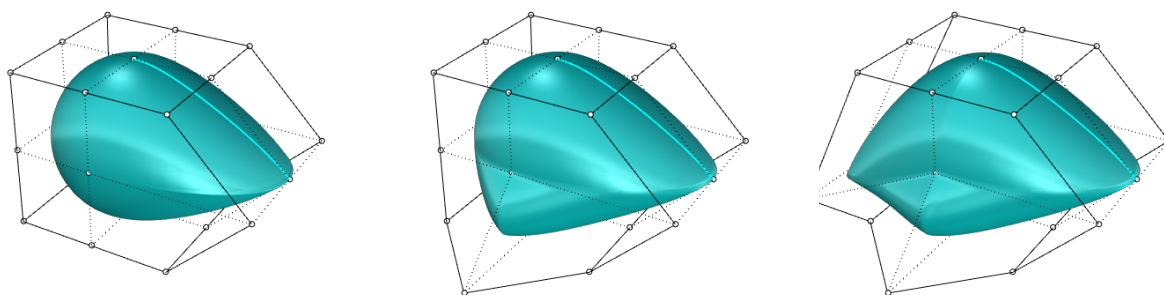
Parameter	Global Optimization	Local Optimization
Cage Type	Bounding Box	Box
Number of Points along X	7	Case Specific (5 to 9)
Number of Points along Y	3	3
Number of Points along Z	4	3
Degree of Points in X	2	2
Degree of Points in Y	2	2
Degree of Points in Z	2	2

1. It is important to note that the type of cage determines its influence over specific regions of the geometry, while the number and degree of points along a given direction affect the smoothness and continuity of the surfaces.
2. A higher number of control points provides greater user flexibility but also increases the number of design parameters, which can complicate the optimization process.
3. For hull forms, it is crucial to respect the symmetry of the hull lines. Control points should be chosen and distributed around the Y-axis (3 points) in such a way that their center always aligns with the line of symmetry.
4. Although the transformations primarily concern the wetted surfaces of the hull, care must be taken to ensure that the modified surfaces are seamlessly integrated with the rest of the hull form.
5. Figure 6 illustrates a spherical object encapsulated within a cubic lattice comprising three control points along each axis. A similar set of transformation were performed in Figure 7 and 8 to demonstrate the effect of Degree of Cage points on the resulting surfaces.

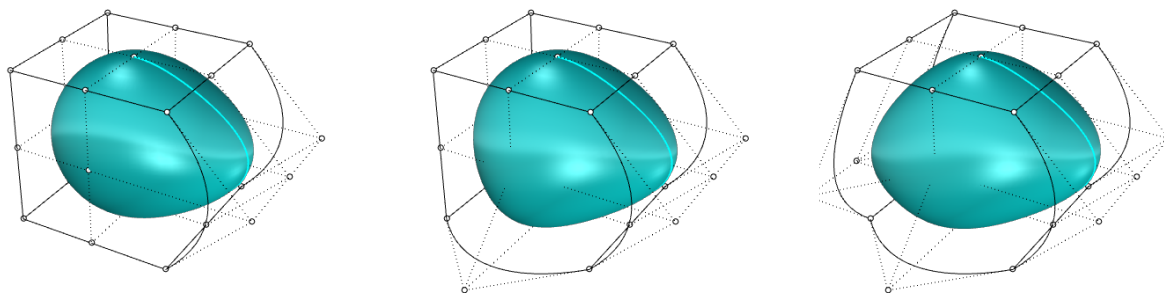




**Figure 6:** Sphere within a cubic lattice Cage



**Figure 7:** Cage Edit with Degree 1

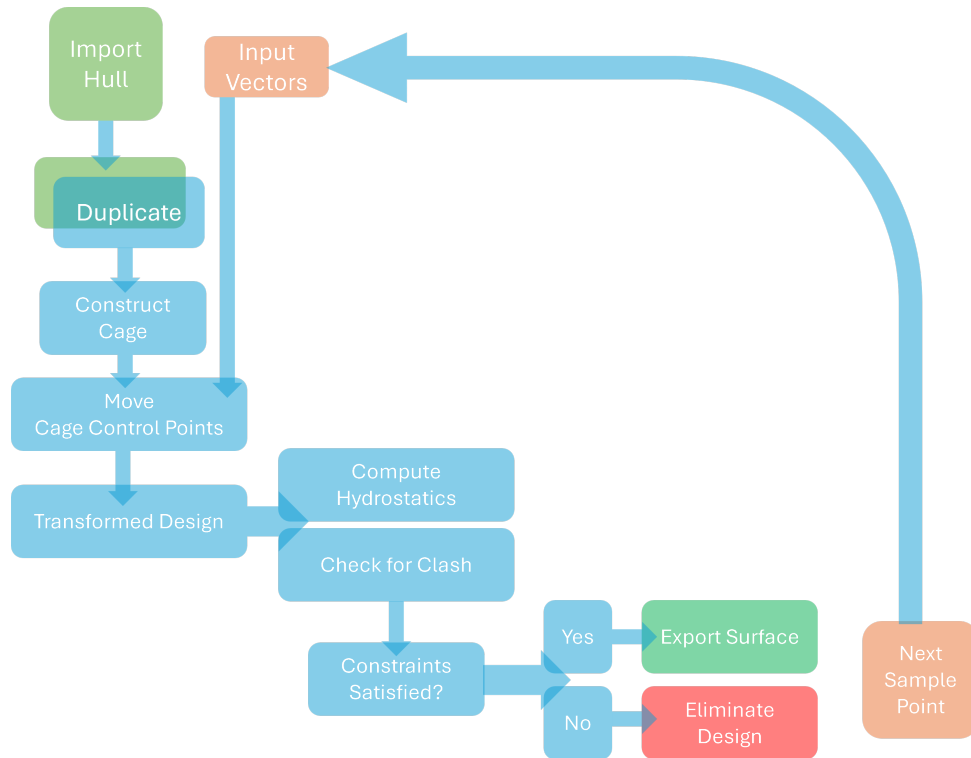


**Figure 8:** Cage Edit with Degree 2

The subsequent sections will elaborate on the implementation of this feature for hull form modifications. Additionally, the concepts of local and global optimization within the FFD framework will be explored, providing a comprehensive approach to shape optimization.

### 4.1.3 Python Script for Rhino

In order to automate the process of cage creation followed by specific perturbations, a Python script was developed. Figure 9 gives an overview of the inner-working of the Rhino Python script. Likewise, Figure 10 shows a folded version of the actual Rhino Python script and the function definitions within this script. Its workflow is as follows:



**Figure 9:** Workflow of Rhino Python Script

1. Import the original hull surface geometry.
2. Validate the input surface for topological integrity, ensuring it is a *closed polysurface* and rectifying any holes or *naked edges*.
3. Generate a duplicate of the hull surface while preserving the original input as a hidden reference.
4. Construct a deformation cage using predefined parameters specified by the user. (using `create_cage_box` function)
5. Execute a series of controlled modifications on the duplicated hull by systematically displacing the cage control points according to predefined

- algorithms. (using `textttlength_param`, `beam_param` `angle_param` functions)
6. Compute the hydrostatic properties at a specified draft for both the original and modified hull forms. (using `'get_hydrostatics'` and `'change_in_hydrostatics'` functions)
  7. Quantify the variations in critical hydrostatic parameters (e.g., Longitudinal Center of Buoyancy, Displacement, Metacentric Height) as defined by the optimization problem constraints.
  8. Perform collision detection between the modified hull and user-defined external objects (e.g., bow thruster tunnels, gearboxes, tanks) to ensure spatial compatibility. (using `'clash_check'` function)
  9. If all constraint criteria are satisfied, export the modified surface in both Rhinoceros (`.3dm`) and Parasolid (`.x_t`) file formats. (using `'export_poly_surf'` function)
  10. Prepare the Parasolid file for subsequent (CFD) analysis using FineMARINE software

```

2   import Rhino
3   import scriptcontext as sc
4   def read_points(file_path):
16  def read_local_points(file_path):
17  def saveopttxt(opttxt):
33  def savetxt(txt, name):
41  def save_summary(txt, name):
42  def loqtxt (string, s):
49  def export_poly_srf(poly_srf_id, name, location, file_extension, rhino version):
57  def get_hydrostatics(save_data, draft, poly_srf_id, tolerance_hydro):
72  def change_in_hydrostatics(original_srf_id, poly_srf_id, draft, s):
73  def create_cage_box(p1,p2,p3,p4,xptct,yptct,zptct,xdeg,ydeg,zdeg):
80  def create_global_cage_box(polysurf_id, xptct,yptct,zptct,xdeg,ydeg,zdeg):
81  def execute_cage_edit(poly_srf_id, cage_id, influence, fall_off_dist = 2):
143  def execute_global_cage_edit(polysurf_id, cage_id):
182  def length_param(cage_ctrl_id, xlim, ylim, zlim, disp):
183  def beam_param(cage_ctrl_id, xlim, ylim, zlim, scale=1):
190  def angle_param(cage_ctrl_id, xlim1, ylim1, zlim1, xlim2, ylim2, zlim2, angle_deg=0, center=[0,0,0]):
191  def get_updated_cage_limits (cage_ctrl_id = None):
198  def clash_check (obj_id, poly_srf_id, clearance):
206  def get_obj_id(obj_filter_index):
207  def cage_pt_sel(cage_id,xlim,ylim,zlim):
211  def move_grips(selected_pts, xform):
212  def main_global (design_id, save_location, cage_only, global_x_loc, b):
226  def main_local (design_id, save_location, cage_only, l, b, s, a):
227  if name == " main ":
251

```

Read Input Points

Export Model

Estimate Hydrostatics

Cage Definition

Displace points along X

Scale points along Y

Rotate points about Y

Physical Constraint

Global Optimization

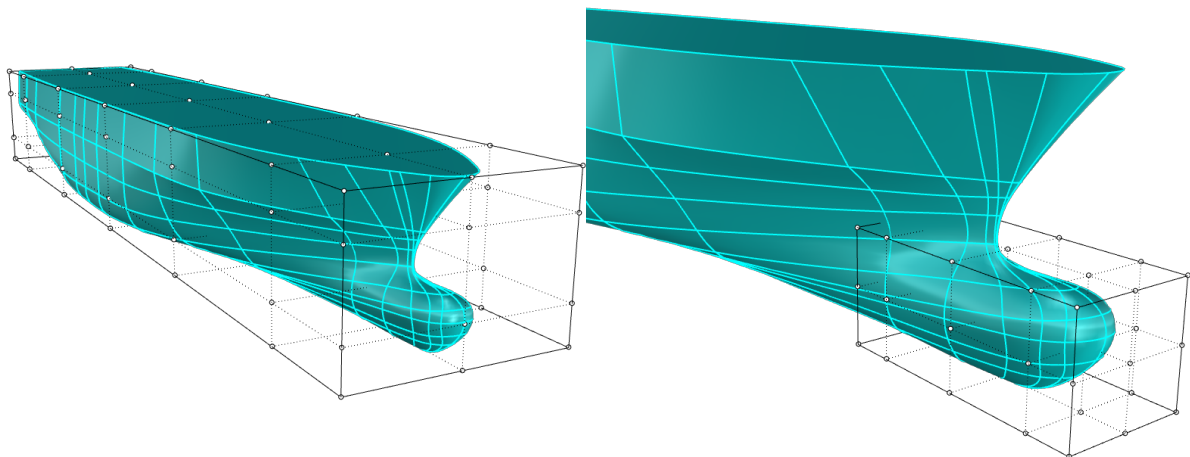
Local Optimization

MAIN function

Figure 10: Definition of Rhino Python Script

#### 4.1.4 Concept of Global & Local Optimization

Global Optimization	Local Optimization
Global optimization involves the entire hull form, treating it as a single polysurface. The cage surrounds the entire hull, and any deformation of the control points results in modifications across the entire surface.	Local optimization targets specific areas, such as appendages or the bulbous bow. It uses a lattice cage surrounding the targeted region, which can vary depending on the vessel and desired outcomes.
The general strategy for global optimization is consistent across most vessel types.	Local optimization requires careful definition of cage control points and their deformation vectors.
Global optimization is driven by a limited number of variables (cage control points), requiring minimal user input to initiate the process.	Local optimization depends on multiple factors and requires careful attention by the user in defining the cage control points and their deformation vectors accurately.
The resulting hydrostatics are highly sensitive to global transformations. Likewise, similar trend was observed for changes in surge resistance following the CFD analysis.	The resulting hydrostatics are less sensitive to local transformations while the sensitivity of surge resistance varied on a case to case basis.



**Figure 11:** Cage for Global Optimization (left) & Local Optimization (right)

Figure 11 illustrates a control cage encompassing the entire hull structure for global optimization, contrasted with a localized cage surrounding only the bulbous bow region for local optimization. The Rhino Python script can adapt to global transformations by switching the respective flag on or off. However, the full extent of these strategies requires further testing on various hull forms.

### 4.2 Fine Marine based Resistance Calculation

Fine Marine, is a commercial computational fluid dynamics (CFD) software primarily used in the marine industry for performing hydrodynamic calculations related to ship resistance, propulsion, seakeeping and multi-phase flow problems.

FineMarine serves as the primary Computational Fluid Dynamics (CFD) engine in the current optimization workflow. The software's Python scripting functionality facilitates the automation of project creation and management processes. Moreover, FineMarine incorporates distributed computing, utilizing a remote High-Performance Computing (HPC) cluster for intensive calculations. While this significantly reduced down the computation time for individual cases, it also adds an additional layer of complexity for the optimization workflow where-by the calculations are performed iteratively in a stochastic manner. This leads to a multi-step process involving mesh generation and computation setup on a local system, followed by data transfer to the HPC cluster for numerical solving, and finally, extraction and post-processing of results on the local device.

FineMarine's modular structure comprises three key components: Hexpress for mesh generation, the FineMarine solver for CFD computations, and CFView for results visualization and analysis. Hexpress also includes advanced algorithms to generate optimized mesh cells, accurately resolving viscous boundary layers and incorporating mesh diffusivity and adaptive refinement techniques for simulations involving moving bodies. These features are especially useful for accurate and reliable surge resistance estimation.

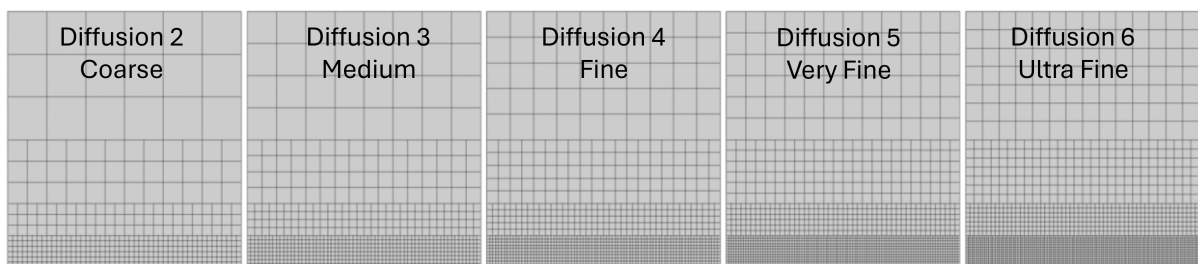
#### 4.2.1 Essential CFD Parameters

This section lists down all the essential CFD parameters concerning the resistance calculation. This involve parameters related to mesh, physical configuration of the system, choice of turbulence model, convergence criteria and some additional numerical parameters. Furthermore, these parameters were selected in close consultation with the experts at the host institution (MAURIC) as well as by extensively referring to the FineMARINE user guide *FINE/Marine Users Guide* [19].

**Mesh Type :** A half body mesh was selected since the body and flow conditions was assumed to remain symmetric throughout the simulation. Furthermore, only bare hull resistance was estimated without considering any effect of hull-propeller interaction.

**Domain Size :** A domain box surrounding the hull body was defined as a function of length between perpendiculars ( $L_{pp}$ ) of the vessel. The dimension of this box was based on the accepted in-house practices as well as FineMARINE recommendations.

**Refinement Level & Diffusion :** These parameters essentially govern the size and quality of the resulting mesh. The refinement level  $n$  divides the initial mesh by a factor of  $2^n$  after  $n$  successive refinements on each surface of the hull. Likewise, the diffusivity factor dictates the transition of mesh from finest cells adjoining hull surfaces to the coarse mesh on the outer domain of the region. Figure 12 illustrates the resulting mesh and its diffusivity for various levels of fineness. A *Fine* mesh with a diffusion of 4 was selected for all the subsequent CFD simulations considering the total computation time and desired accuracy of results.



**Figure 12:** Mesh Refinement & Diffusion

**Viscous Layers :** The primary purpose of adding strips of viscous layers is to capture the boundary layer effects of the flow. This is especially relevant for resistance calculation wherein the velocity gradient is steep and shear stresses are significant. While the underlying theory of viscous layers calculations is been widely discussed in academic literature, the focus here is to interpret FineMARINE's approach of inserting viscous layers within a mesh. In this regard, HEXPRESS module of FineMARINE first inserts cells of a very large

aspect ratio near the hull surface followed by successive subdivision of these cells adjacent to the walls (hull surfaces). This is in contrast to other techniques wherein layers are added by means of extrusion of the wall surfaces in normal direction. Furthermore, HEXPRESS estimates the viscous layer parameters through the following relations

$$y_{\text{wall}} = 6 \left( \frac{V_{\text{ref}}}{v} \right)^{-\frac{7}{8}} \left( \frac{L_{\text{ref}}}{2} \right)^{\frac{1}{8}} y^+ \dots \quad (5)$$

$$y^+ = \max(y_{\text{min}}^+, \min(30 + \frac{(Re - e^6) * 270}{e^9}, y_{\text{max}}^+)) \quad (6)$$

$y_{\text{wall}}$	First layer thickness	m
$V_{\text{ref}}$	Reference velocity of the flow	m/s
$v$	Kinematic viscosity of the fluid	m <sup>2</sup> /s
$L_{\text{ref}}$	Reference length (Length of waterline)	m
$Re$	Reynolds number	-
$y^+$	Non-dimensional wall distance ( $y_{\text{min}}^+ = 50$ , $y_{\text{max}}^+ = 300$ )	-

**Body Motion Dof:** Any given floating body motion can be described by its 6 degrees of freedom. In case of a ship moving with uniform velocity in calm waters, it undergoes a steady state motion where-by some of the degrees of freedom can be neglected. Thus, **surge motion** is externally applied (**imposed**) on the vessel whereas the resulting **heave** and **pitch** motions are evaluated by the solver (**solved**). Moreover, **sway**, **roll**, **yaw** motions are neglected (**fixed**) due to the symmetric nature of the flow and body.

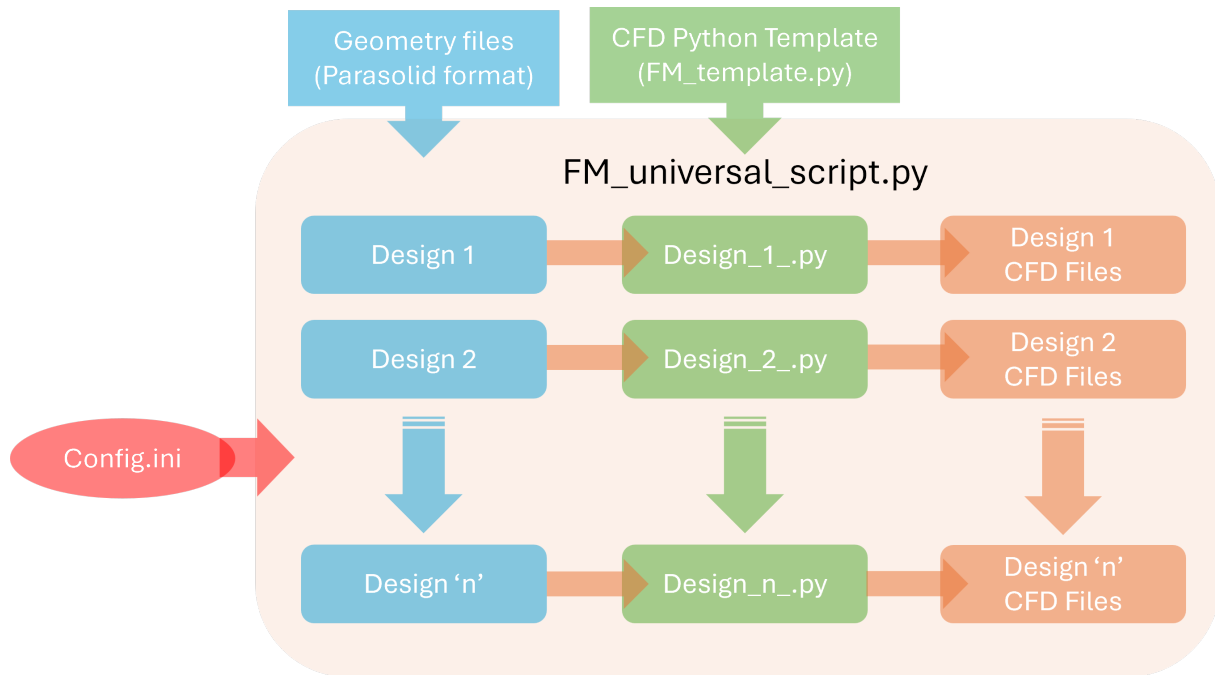
**Body Motion Law:** Since only surge motion is imposed on the body, it is important to also define a motion law which dictates the manner in which the force is applied on the body as a function of time. A *1/2 sinusoidal ramp* was chosen to define the change in velocity profile of the vessel from zero to operational speed ( $V_{\text{ref}}$ ) in a defined time interval (20 seconds in most cases). This selection was based on standard in-house practices followed by the host institution (MAURIC).

**Table 4:** CFD Parameters

Mesh Type	Half Body
Domain size	$5L_{pp}$ (along X axis) $1.5L_{pp}$ (along Y axis) $2L_{pp}$ (along Z axis)
Global Refinement Diffusion	4
Maximum number of refinements	10
Refinement of Wetted Surfaces	7
Refinement of Free surface	7
Viscous first Layer Thickness	Case Specific
Viscous Layer Stretching Ratio	1.2
Number of Viscous Layers	15 to 35
Y+ value for Viscous layer	Case Specific
Turbulence Model	K-Omega (SST Menter)
Body Motion Dof	surge(imposed) sway(fixed) heave(solved) roll(fixed) pitch(solved) yaw(fixed)
Body Motion Law	1/sinusoidal ramp
External Forces	Drag-based wrench by propeller
Adaptive grid refinement	Activated
Number of time steps	2000
Time step value	0.04 to 0.08



## 4.2.2 Automation of CFD



**Figure 13:** Workflow of CFD Automation

The workflow begins with the preparation of a CFD setup file in FineMARINE, defining essential parameters and exporting it as a Python script (*FM\_template.py*). This script serves as a template for individual designs.

A universal configuration file, *config.ini*, is established to manage all user input parameters related to the entire workflow. This file serves as the primary interface for users to set up the workflow parameters. The *FM\_universal\_script.py* was developed to automate the CFD workflow. This script performs the following functions:

1. It accepts geometry files in parasolid format as input.
2. Based on the input geometry, it makes necessary adjustments to the *FM\_template.py* file. These adjustments make sure that the hydrostatic properties and input-output file locations are updated with each design.
3. It then generates design-specific script (*Design\_1\_.py*) for each input geometry which serves as an executable script for the FineMARINE to generate respective CFD files.

The *FM\_universal\_script.py* executes in a loop, processing multiple design files ('n' in total). For each iteration:

1. It reads the geometry file and configuration parameters.
2. It modifies the template file according to the specific design requirements.
3. It generates a unique *Design\_n\_.py* script.
4. This script is then automatically executed in FineMARINE, resulting in the creation of corresponding CFD files (*Design 'n' CFD Files*).

The process repeats for each design, ensuring consistent and automated CFD analysis of multiple geometries in series. An overview of this process is described in Figure 13

### 4.3 Python based Integration Framework

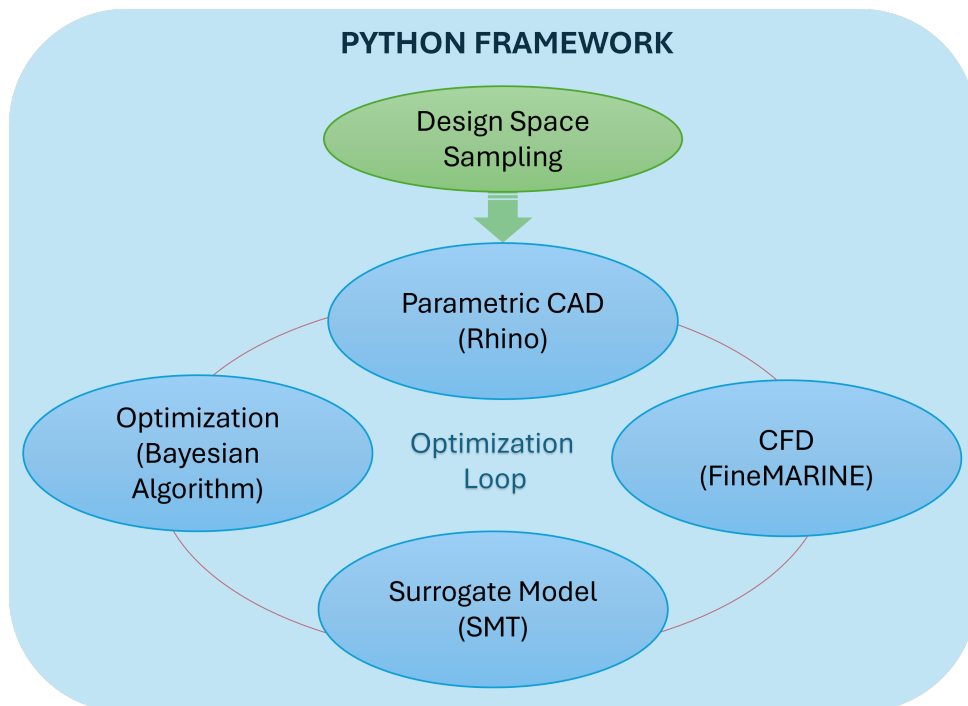
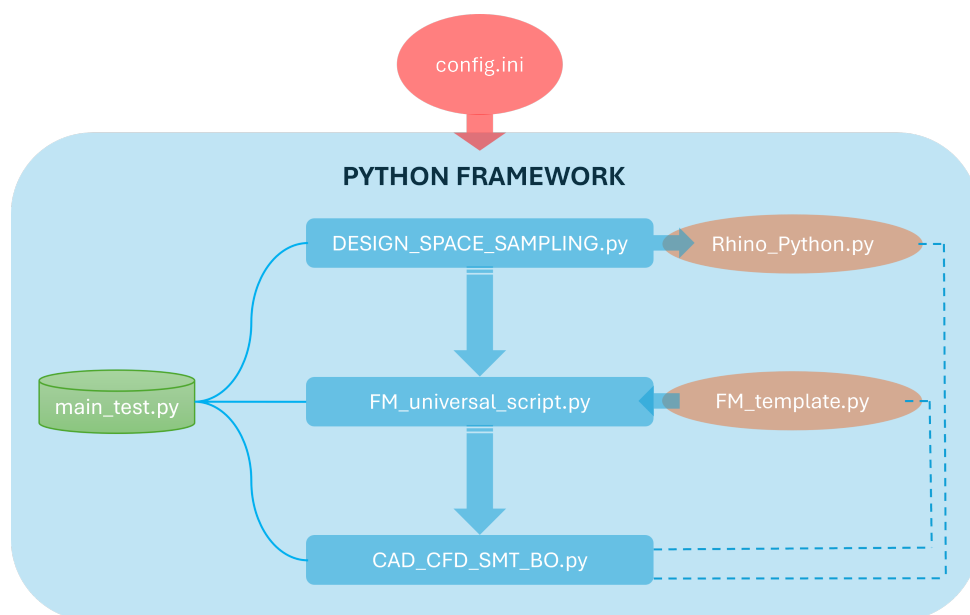


Figure 14: Optimization Workflow

- The core of the optimization workflow is built upon a Python-based integration framework.

- This framework seamlessly integrates individual applications such as Rhino, FinemARINE, Surrogate Modeling and the Bayesian Optimization toolbox within a unified architecture.
- The framework (explained in Figure 15) is structured around multiple key scripts, including:
  1. DESIGN\_SPACE\_SAMPLING.py
  2. FM\_universal\_script.py (explained in Figure 13)
  3. CAD\_CFD\_SMT\_BO.py (explained in Figure 16)
- To ensure efficient compilation, smooth data transfer, and seamless execution of the workflow, all scripts import necessary functions and libraries from a centralized Python function repository defined in main\_test.py.
- Robust control statements are also incorporated at each step of the workflow to effectively manage potential errors during script execution.
- Additionally, relevant messages are printed to the main terminal window to guide the user through subsequent steps in the process.
- The input file config.ini serves as the primary configuration file, where users define critical workflow parameters.

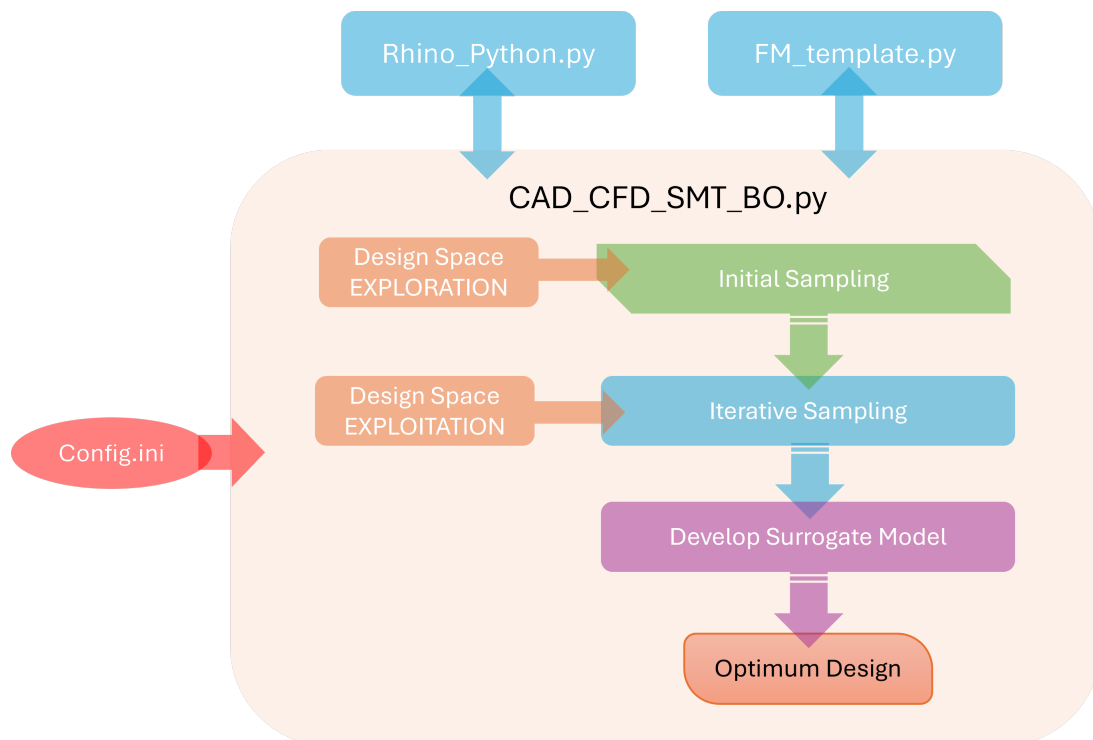


**Figure 15:** Python Framework Architecture

Figure 15 illustrates the inner working and an overall architecture of the Python based integration framework. While it inculcates many additional subroutines and processes, the subsequent points elaborate the necessary individual steps of this framework in a simplified and coherent manner.

1. The user first defines all the essential and relevant parameters in the `config.ini` file, followed by the execution of the `DESIGN_SPACE_SAMPLING.py` script.
2. The `DESIGN_SPACE_SAMPLING.py` script generates the initial sampling points of the chosen design parameters using the Surrogate Modeling Toolbox, based on the Latin Hypercube Sampling method. These design parameters can either be explicitly defined by the user or chosen from the following examples:
  - (a) Scaling parameters (Ba | Bb) for Global Optimization
  - (b) Bulb parameters (Length (L) | Beam (B) | Sharpness (S) | Angle (A)) for Local Optimization
  - (c) Any other user-defined parameter
3. The numeric values of these points are stored in a `sample_pts.txt` file. The `Rhino_Python.py` script reads these points, performs the corresponding CAD modifications, and exports the model. The workflow is extensively explained in Figure 9.
4. Once the design files (CAD models) are generated, they are iteratively passed to the `FM_universal_script.py` script, along with a CFD template file (`FM_template.py`), to generate FineMARINE files specifically adapted to each design. This process is elaborated in Figure 13.
5. Finally, the `CAD_CFD_SMT_BO.py` script is executed, performing tasks in the following order. Its workflow is depicted in Figure 16.
  - (a) Transferring the CFD files to a remote cluster, triggering the solver, and extracting the results. This marks the end of the exploration phase of the optimization and begins the iterative design space exploitation.

- (b) The script is coupled with the Bayesian Optimization toolbox, which stochastically suggests new design points based on the results obtained from the previous set of points, triggering the design space exploitation phase.
- (c) Once the maximum number of iterations, as set by the user, is reached, a Surrogate model is created using the data from the sampling points and their corresponding CFD results.
- (d) Finally, the optimum point of this surrogate model is evaluated. This point represents the design parameters (e.g., Ba & Bb) corresponding to the most optimized design.



**Figure 16:** Surrogate Modeling and Bayesian Optimization workflow

The table presented in section 4.4 shows the layout of the main user input file 'config.ini'. This file serves as the primary repository of all the essential optimization parameter given or defined by the user. In addition, all the scripts and programs associated with the framework receives their input argument by reading the data or values from config.ini.

## 4.4 Main user input file layout

Table 5: config.ini - input file layout

<b>[GENERAL]</b>	
CFD_FOLDER_NAME	= CFD (Folder name to save FineMARINE files)
number_of_samples	= 20 (Sampling points for Design Space Exploration)
n_iter	= 10 (Sampling points for Design Space Exploitation)
random_state	= 7 (Seeder to reproduce Sampling Points)
fm_summary_path	= D:\PATH\TO\FM_SUMMARY.csv
fm_iterative_summary_path	= D:\PATH\TO\FM_LHS_BO_SUMMARY.csv
bayesian_optimization_log	= D:\PATH\TO\BO_LOGS.log
avg_of_last_x_percent	= 10 (Resistance as Avg. of last 10% time-steps)
<b>[LHS]</b>	
pbounds	= { `Ba': (0.8, 1.2), `Bb': (0.7, 1.4) } (Upper & Lower Limits of Design Parameters)
default_limits	= { `Ba': 1, `Bb': 1 } (Default value of Design Parameters)
<b>[VESSEL]</b>	
VESSEL_NAME	= SXMSV (Name of reference vessel)
L_pp	= 82 (Length of Waterline in meters)
v_ref	= 7.717 (Reference velocity in m/s)
DWL	= 3 (Design draft from keel in meters)
Z_COG	= 4.9 (Vertical CoG from keel in meters)
<b>[FM]</b>	
domain_coeff	= [-3, 2, 0, 1.5, -1.5, 0.5]
adaptive_ref_coeff	= [-0.475, 1.05, 0, 0.3, -100, 100, -0.475, 1.05, 0, 0.3, -100, 100, -0.95, 100, 0, 0.95, -100, 100 ]
initial_cartesian_mesh	= [20, 6, 6] (Division of Initial Cartesian Mesh)
refinement_increment	= 0 (Increase existing mesh refinement level) (for Multi-fidelity simulation))
<b>[PATHS]</b>	
RHINO_geometry_file_path	= D:\PATH\TO\rhino_geometry_file.3dm
RHINO_python_script_path	= D:\PATH\TO\rhino_python_script.py
RHINO_exe_location	= C:\PATH\TO\rhino_executable.exe
RHINO_bat_file_location	= D:\PATH\TO\rhino_bat_file.bat
PBS_template_loc	= D:\PATH\TO\PBS_template_for_HPC.pbs
FM_template_path	= D:\PATH\TO\FineMARINE_Python_Template.py
FM_universal_script_path	= D:\PATH\TO\FM_universal_script.py
FM_fun_def_path	= D:\PATH\TO\main_test.py
FINEMARINE_exe_location	= C:\PATH\TO\FineMARINE_executable.exe
sampling_points_txt_file	= D:\PATH\TO\samples_pts.txt
<b>[HPC]</b>	
host	= 192.168.000.000 (I.P address of remote cluster)
port	= 22 (Default Value)
username	= ABC (Username ID of cluster)
password	= **** (Password)
HPC_remote_dir	= /PATH/TO/CLUSTER/DIRECTORY
hpc_launch_intel_pbs_location	= /PATH/TO/launch_Intel_PBS_v121.py
check_HPC_status_interval	= 600 (seconds)
<b>[ADDITIONAL]</b>	
fm_version	= 121
DELETE_bxx	= 1 (or True)

## 5 TEST CASE I : PATROL VESSEL

Following the development of the Python based optimization framework, it was tested on multiple vessels for both the local and global optimization. This section presents a use-case scenario of the initial setup and implementation of the optimization workflow. Furthermore, it must be noted that running these test cases serves an additional opportunity to encounter and thus fix any underlying bugs which may have otherwise went unnoticed during the initial development of various scrips and modules which collectively form the framework.

The present test case is based on fore part and bulb modification of a 54 meter patrol vessel. Moreover, **for reasons of confidentiality, some aspects of the vessel might be indicative in nature or omitted altogether.** However, key points related to the optimization results will be discussed with emphasis on potential modifications to the framework and fine-tuning of surrogate models.

### 5.1 Optimization Workflow & Setup

The optimization process for the vessel was conducted independently on two distinct regions: the fore part and the bulbous bow. Initially, a global cage was employed to optimize the fore part, followed by a local cage focused on the bulbous bow area. In both cases, the primary objective was to minimize surge resistance, with constraints being set to restrict the change in longitudinal center of buoyancy (LCB) and volumetric displacement ( $\nabla$ ) to 1.5%. Additionally, physical constraints were imposed to avoid clashes of modified hull surfaces with the two bow thrusters. The global optimization utilized two design parameters,  $B_a$  and  $B_b$ , which are scaling factors for specific cage points in the fore part. In contrast, the local optimization for the bulbous bow was based on four design parameters: length, breadth, sharpness, and angle of the protruding bulb.

### 5.1.1 Vessel Characteristics

**Table 6:** Characteristics of a Generic Patrol Vessel

Name	Generic Patrol Vessel
Length Overall	54 m
Beam	11.7 m
Height	6.7 m
Draft	3 m
Block Coefficient ( $C_B$ )	0.449
Displacement	810 tons
Speed	13 knots
Propulsion	Single Screw type
Thruster	2x Bow Thrusters
Bulb	'O' type

### 5.1.2 Objective function & constraints

**Table 7:** Optimization Table for Minimizing Surge Resistance

<b>Objective Function</b>	Minimize Surge Resistance at 13 knots
<b>Constraints</b>	
Change in LCB	$\pm 1.50\%$
Change in Displacement	$\pm 1.50\%$
Clashing Objects	2x Bow Thruster Tunnels

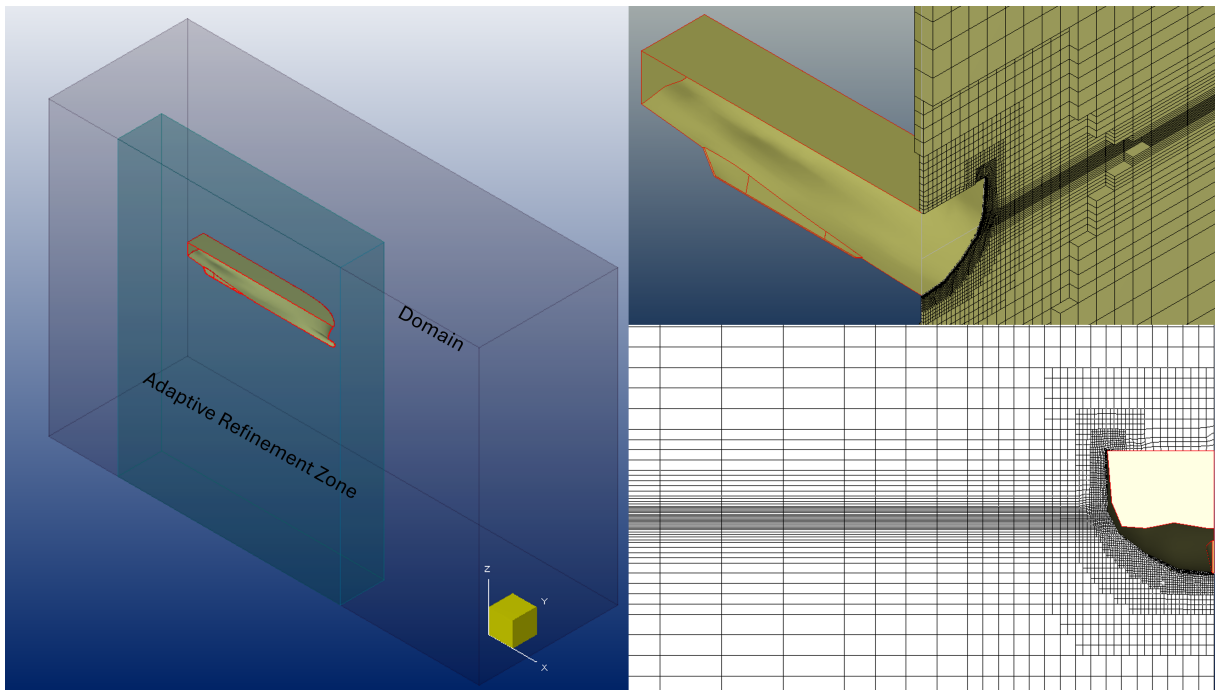
### 5.1.3 CFD setup

**Domain Size:** The CFD domain was defined as a function of  $L_{pp}$  of the vessel. This parameter can be defined in terms of list of coefficients under the 'domain\_coeff' & adaptive\_ref\_coeff parameters in *config.ini* file as shown in Table 5. In the present case, the dimensions are as follows:

Dimension of Outer Domain Zone :  $270 \times 81 \times 108$  m

Dimension of Adaptive Refinement Zone :  $82.35 \times 82.35 \times 102.6$  m





**Figure 17:** Patrol Vessel CFD Domain

**Mesh Quality:** A maximum refinement of 10 was selected along with a global diffusion of the order 4 for gradual transition of cell size within the domain. These parameters corresponds to a ‘Fine’ level of mesh in Figure 12 resulting in an overall mesh size of about 2.4 million cells.

**Body Motion Law & Time Steps:** The CFD simulation was performed for a total of 2000 time steps of 0.04 seconds each resulting in a total simulation time of 80 seconds. Moreover, the velocity of 13 knots (6.688 m/s) was gradually imposed from 0 to 20 seconds based on a  $1/2$  sinusoidal ramp profile.

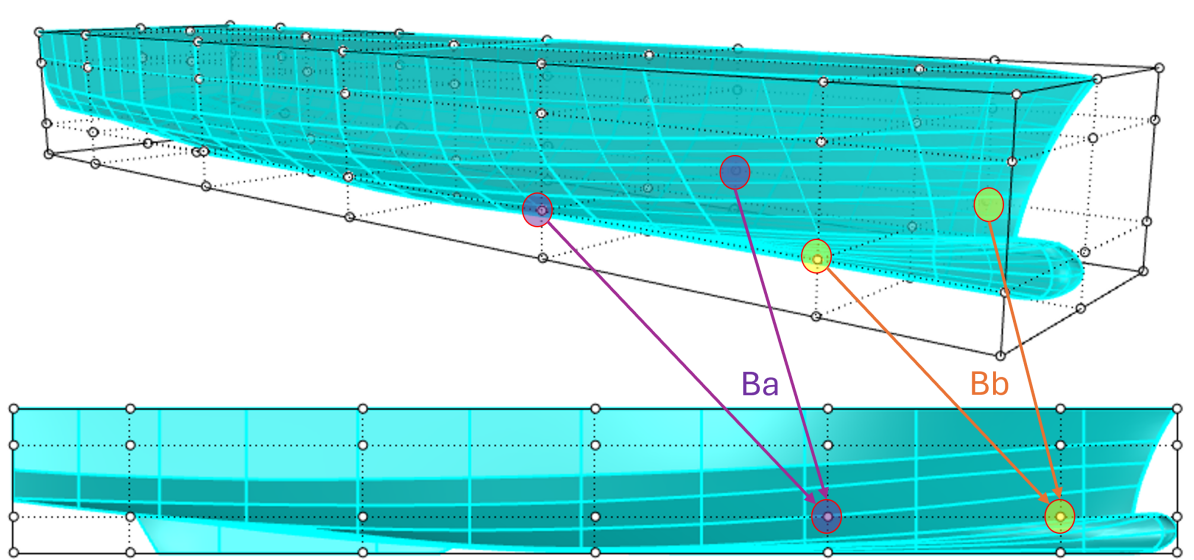
**Additional Parameters:** While all the CFD parameters have not been exhaustively mentioned, a brief summary of most essential parameters can be found in Table 8.

**Table 8:** Patrol Vessel CFD Parameters

Mesh Type	Half Body
Total Cells	$2.388 \times 10^6$
Domain size	270 m (X axis) 81 m (Y axis) 108 m (Z axis)
Global Refinement Diffusion	4
Maximum Number of Refinements	10
Refinement of Wetted Surfaces	7
Refinement of Free Surface	7
Viscous First Layer Thickness ( $Y_{wall}$ )	0.001452 m
$Y^+$	138
Viscous Layer Stretching Ratio	1.2
Number of Viscous Layers	13 to 17
Turbulence Model	K-Omega (SST Menter)
Body Motion DOF	Surge (imposed) Sway (fixed) Heave (solved) Roll (fixed) Pitch (solved) Yaw (fixed)
Body Motion Law	1/sinusoidal ramp
External Forces	Drag-based wrench by propeller
Adaptive Grid Refinement	Activated
Number of Time Steps	2000
Time Step Value	0.04 s

## 5.2 Optimization of Fore Part

### 5.2.1 Global cage modifications & Initial Sampling



**Figure 18:** Global Design Parameters (Ba,Bb) for Patrol Vessel

The hull model was encapsulated within a cage defined by the parameters mentioned in Table 9. These parameters were based on the following considerations

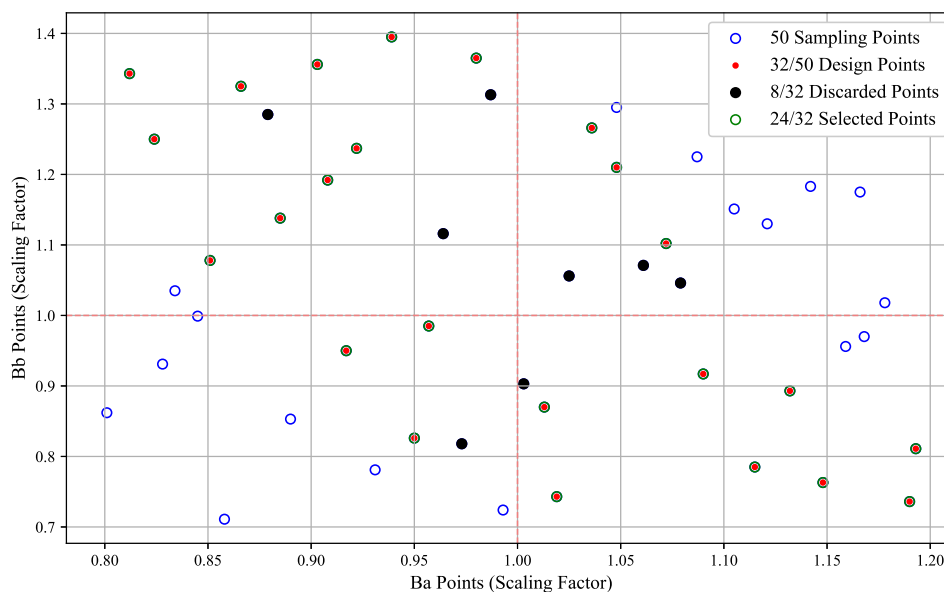
- There must be at-least one slice of cage points below the design water line.
- The points must be evenly distributed about the plane of symmetry (*XZ Plane*).
- The hard chine due to integration of Bulb with hull surface must not be modified.
- Only the fore part of the vessel must be modified while the fall off on the aft region must be restricted to close tolerances.
- The modified surface must satisfy at-least  $C^1$  continuity of tangency.

Thus, four set of cage points were selected and their modifications were governed by two scaling factors Ba & Bb. Figure 18 illustrates the selection of these points and their associated scaling factors.

**Table 9:** Global Cage Edit Parameters for Patrol Vessel

Cage Type	Bounding Box
Number of Points along X	7
Number of Points along Y	3
Number of Points along Z	4
Degree of Points in X	2
Degree of Points in Y	2
Degree of Points in Z	2

Following the definition of the cage, cage points and modification strategy, an initial sampling set of fifty unique samples were derived using the Latin Hypercube Sampling method provided by the Surrogate Modeling Toolbox. This was performed by executing the `DESIGN_SPACE_SAMPLING.py` script.

**Figure 19:** Design Space Sampling for Global Optimization of Patrol Vessel

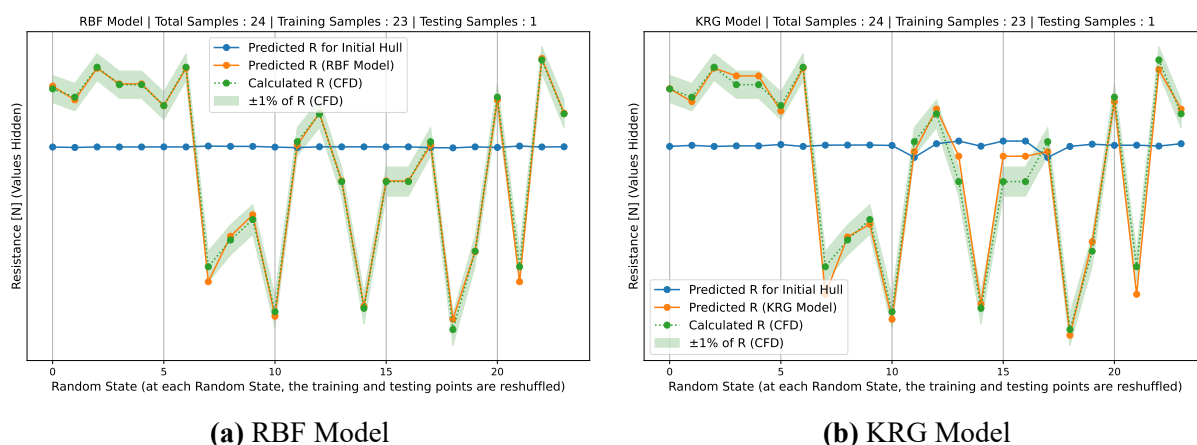
The 50 initial samples were further filtered where 18 were eliminated by the `Rhino_Python.py` script for not satisfying the design constraints and 8 additional points were manually discarded by the user based on design judgement. This step of manual filtering of points could however be eliminated in subsequent test cases to let the optimizer explore an entire spectrum of design space without any user induced bias.

Finally, 24 points were selected and subjected to subsequent steps of the

workflow. In addition, if the user is not satisfied by the resulting set of points, then they can reproduce the initial sampling set with a different random state (in `config.ini` file) and thus repeat the previous steps.

The Python Optimization Framework is then allowed to run independently and uninterrupted for 50 to 60 hours to perform the necessary resistance calculations where by the optimizer undergoes an exploration phase based on the initial sampling set followed by an adaptive sampling phase where-in the optimizer augments its search towards region exhibiting promising results.

After the optimizer reaches the maximum iterations as set by the user (in `config.ini` file), a surrogate model is created based on the data generated from the initial and adaptive sampling points. Finally, the maxima (since resistance values are negative) of this surrogate model is evaluated by another optimizer. This maxima represents the most optimum set of scaling factors ( $B_a, B_b$ ) yielding least absolute resistance.



**Figure 20:** Surrogate Model Comparison for Patrol Vessel

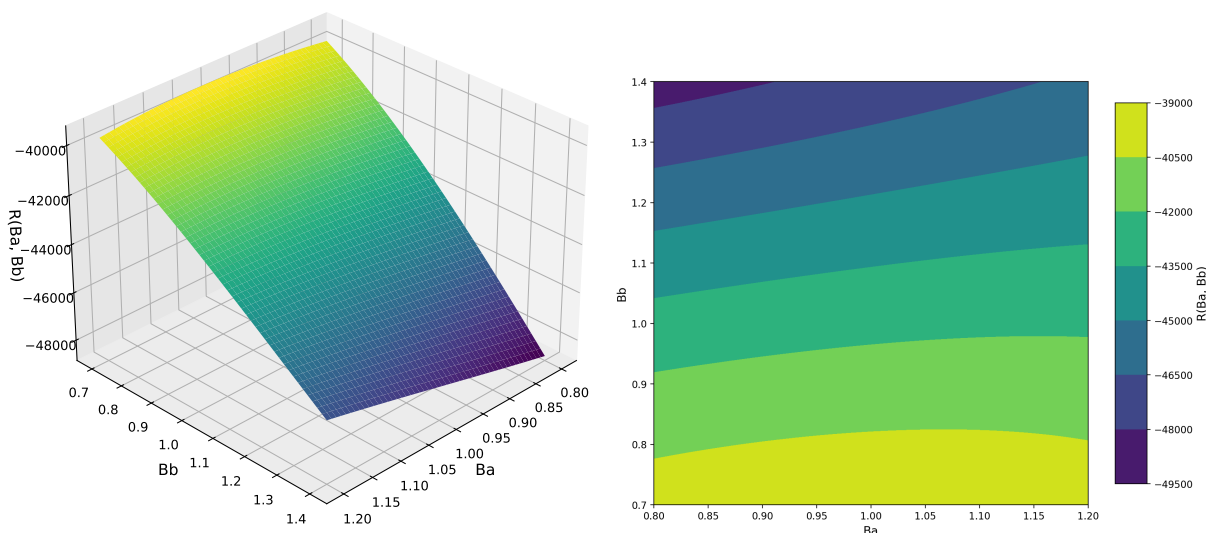
Figure 20 compares the accuracy of two surrogate models i.e. Radial Basis Functions (RBF) & Kriging (KRG) model in terms of their ability to predict the resistance values. Furthermore, **the reshuffling of the points is part of the strategy followed to train and validate the model. This was illustrated in Figure 4.** It can be observed that the RBF model delivers consistent results with the predicted resistance values being within 1% tolerance of the actual calculated value. Likewise, the Kriging model deviates slightly at some points but remains within the set tolerance for majority of points. In both cases, the averaging the results eventually yields a mean value which strongly co-relates with the actual

CFD value. This is shown by the ‘Predicted Resistance (R) for Initial Hull’ whose value falls within a close range of the actual value.

This study was also conducted on various other surrogate models by training them with the same set of data. The reader is encouraged to refer to Appendix A [7] for additional information.

### 5.2.2 Results & comparison

Figure 21 maps the entire design space which is a function of the two scaling factors (Ba & Bb). The selection of these two design parameters against a single objective ( $R_{(Ba, Bb)}$ ) enables us to visualize a three dimensional design space which was mapped using the surrogate model as its mapping function. However, in cases exhibiting more than two design variables would result in a multi-dimensional design space which could not be visualized in the same manner.



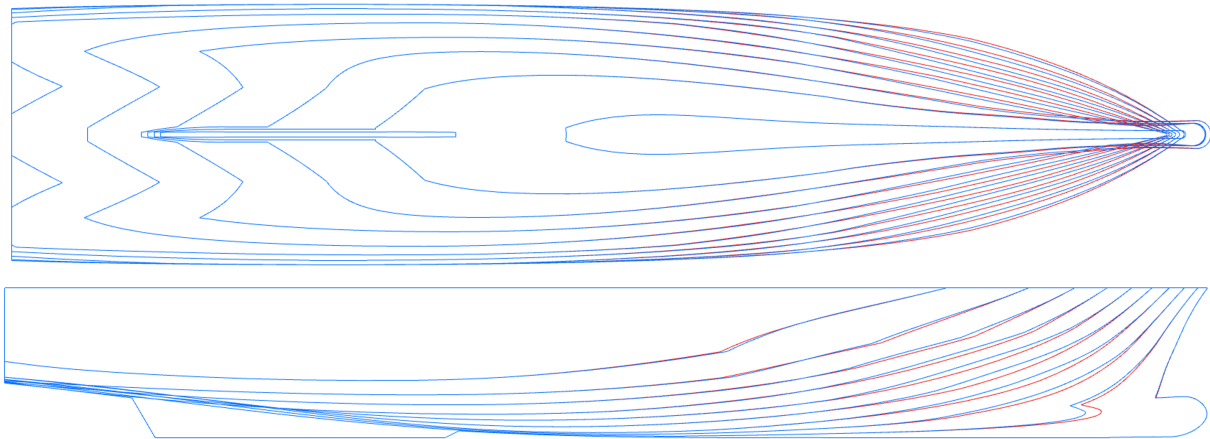
**Figure 21:** Design Space Mapping and Contour for Global Optimization of Patrol Vessel

The maxima of this design space map is evaluated by the optimizer and the results are summarized in Table 10 and for reasons of confidentiality, the author refrains from disclosing the exact resistance values of the concerned vessel. In addition, Figure 22 and Figure 23 draws a comparison of the hull lines between the initial design and final design. Since only fore part of the hull form was modified, the hull lines tend to coincide each other as we gradually move towards

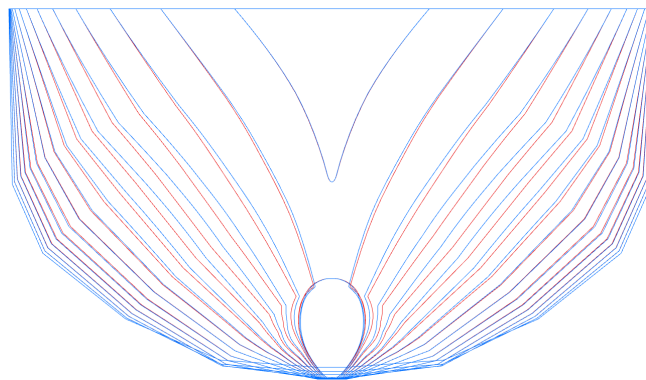
the aft region. This ensures a smooth transition of modifications across the hull form.

**Table 10:** Global Optimization results for Patrol Vessel

	Ba	Bb	Change in Resistance
Initial Design	1	1	0
Design 1	1.030	0.700	-7.59%
Design 2	1.019	0.743	-6.71%

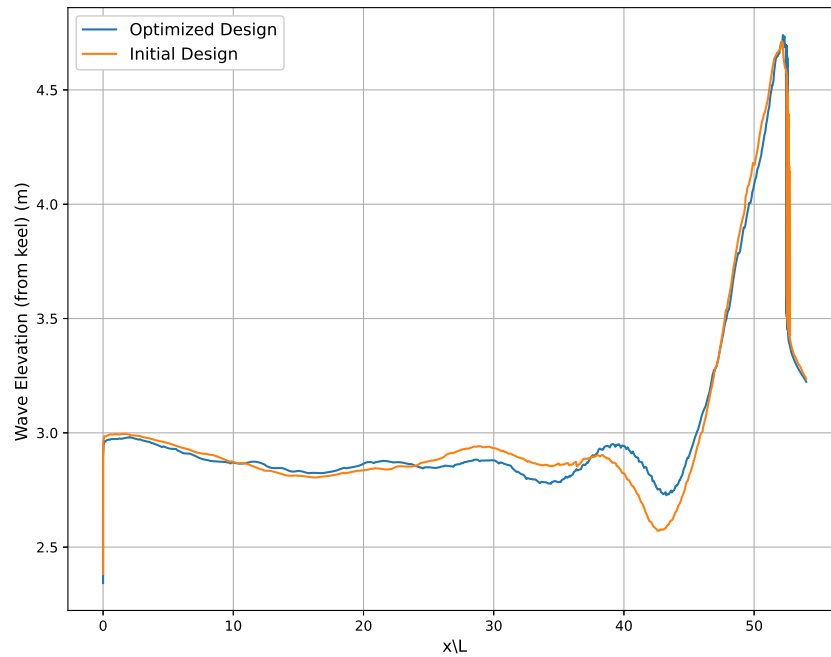


**Figure 22:** Patrol Vessel Hull Lines (INITIAL & MODIFIED)



**Figure 23:** Patrol Vessel Section Lines (INITIAL & MODIFIED)

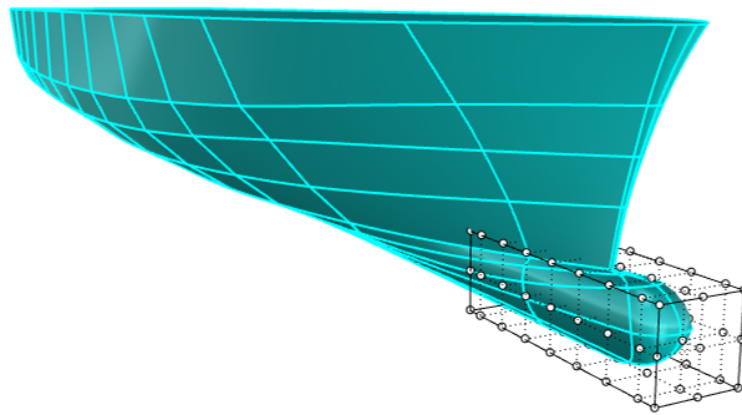
Moreover, one can also observe the wave elevation plot in Figure 24. It must be noted that the elevation is measured from the keel of the vessel for an operational draft of 3 meters at 13 knots.



**Figure 24:** Patrol Vessel Global Wave Elevation Comparison

### 5.3 Optimization of Bulbous Bow

#### 5.3.1 Local cage modifications & Initial Sampling

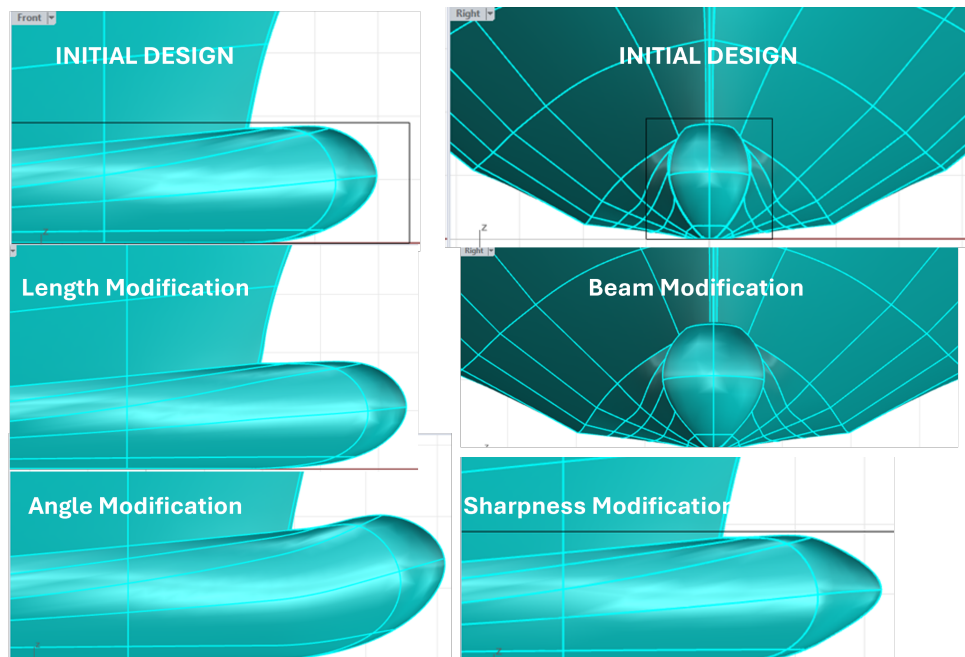


**Figure 25:** Bulb Cage for Local Optimization of Patrol Vessel

- A local cage surrounding the Bulb region of the bow as shown in Figure 25 was constructed with the cage parameters as mentioned in Table 11. A specific set of these cage points were carefully selected and displaced/scaled to perform desired modifications of the bulb.
- Figure 26 illustrates the Length elongation ( $L$ ), Beam scaling ( $B$ ) as well



as Angle (A) and Sharpness (S) modifications applied to the bulb. Thus, these four type of modifications were chosen as design parameters for optimization of bulbous bow. Likewise, the relative magnitude of these parameters in terms of their numerical values were treated as sampling points.



**Figure 26:** Illustrations of Bulb Modifications

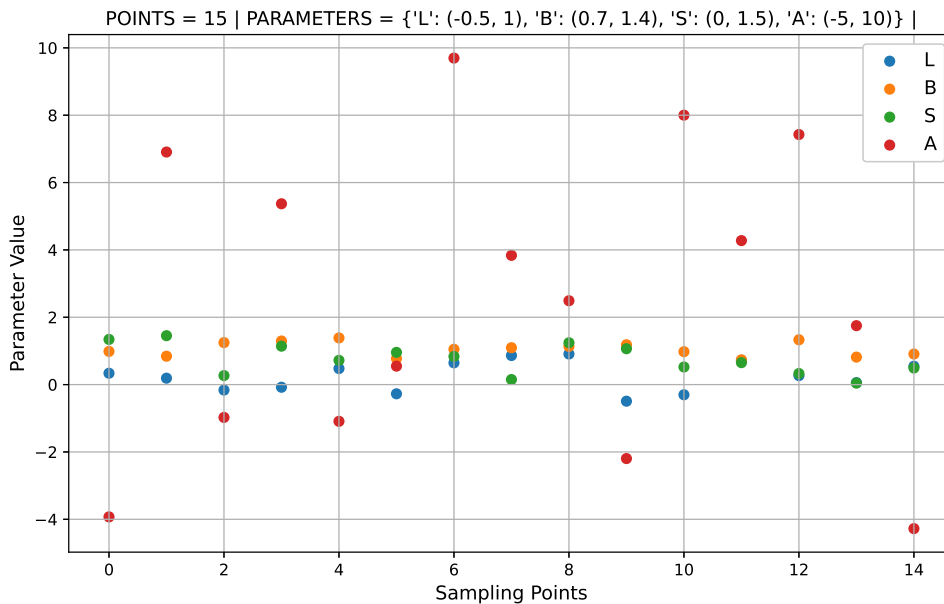
- The length parameter (L) displaces the selected cage points along the longitudinal axis. Thus it can vary from 0 (no change) to any positive real number proportional to the desired elongation.
- The beam parameter (B) scales the cage points symmetrically on either side of *XZ Plane*. Thus it can vary from 0.7 (scaling inwards) to 1.4 (scaling outwards). Moreover, a scale factor of 1 represents No scaling in which case, there is no modification of the bulb beam.
- The sharpness parameter (S) performs in the similar manner as the length parameter except that it displaces a different set of cage points. Thus, this too can vary from 0 (no change) to a value proportional to the sharpness.
- Finally, the angle parameter (A) rotates the bulb surfaces about the Y-axis using the hull-bulb intersection as the point of rotation. It accepts values in

degrees representing the angle of tilt. For practical reasons, the acceptable range of values was restricted to  $-5^\circ$  to  $10^\circ$ .

**Table 11:** Local Cage Edit Parameters for Patrol Vessel

Cage Type	Bounding Box
Number of Points along X	9
Number of Points along Y	3
Number of Points along Z	3
Degree of Points in X	2
Degree of Points in Y	2
Degree of Points in Z	2

- Furthermore, the automated selection and transformation of the cage points was scripted within the `Rhino_Python.py` script along with necessary function definitions to perform these modifications. Figure 10 gives a brief overview of this script and its functions.



**Figure 27:** Design Space Sampling for Local Optimization of Patrol Vessel

- A sampling set was generated for four design parameters as mentioned earlier. The limits of these parameters were set based on a preliminary analysis of bulb shapes for various values of the parameters.

- A total of 15 initial samples were drawn out of which some were eliminated by the script for not satisfying the design constraint. It must be noted that the primary cause of elimination was the physical constraint (2x Bow Thruster Tunnels) rather than hydrostatics. Since the volumetric change of the entire hull due to local modifications is insignificantly small, the hydrostatic variables remain almost similar to the initial design.
- The subsequent steps are similar to the ones followed in the previous case of global optimization and thus their repeated explanation was deemed unnecessary.

### 5.3.2 Results & Comparison

- Figure 28 compares two surrogate models trained against four set of design parameters (L | B | S | A) and a single objective function ( $R_{(L,B,S,A)}$ ). It can be observed that the Kriging model performs better in local optimization which shows better suitability of Kriging model for higher dimensional optimization problems. In contrast, RBF exhibits good performance with lesser design variables and begins to deviate as the number of variables increases. However, this deviation is encountered at only a few data points and does not over-shadow the overall mean value of the predictions.
- Unlike the earlier case of global optimization, a design map cannot be visualized here due to multi-dimensional nature of this problem. However, the optimizer can evaluate the optimum point yielding the absolute minimum objective function  $|R_{(L,B,S,A)}|$ . Moreover, it was realized that the optimizer can converge to different points based on the choice of surrogate model chosen to map the design space. This suggests that although the two surrogate models could predicted similar mean values, they could inherently be different in terms of exhibiting any local maxima or minima. This is a critical aspect of the optimization workflow and the user must prudently compare the suggested optimum value by multiple models before concluding to a single design.

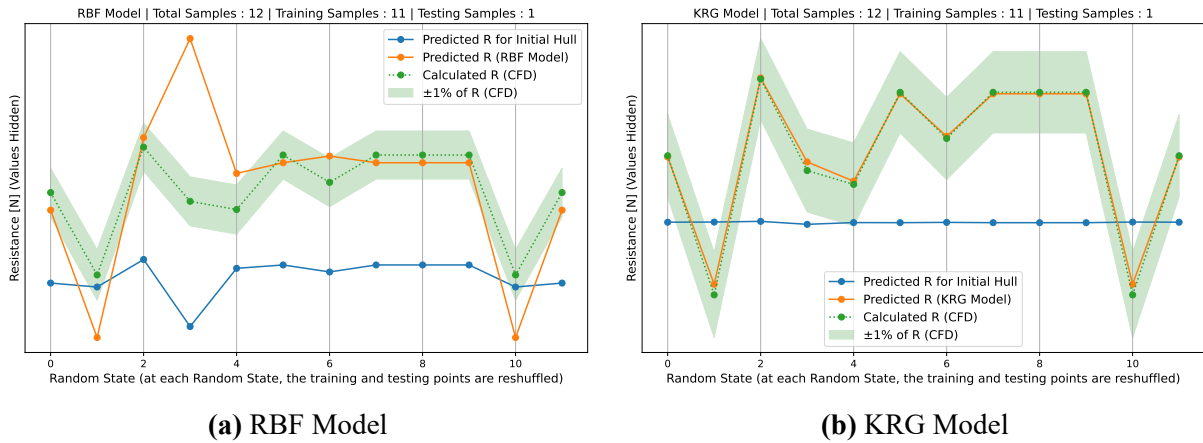
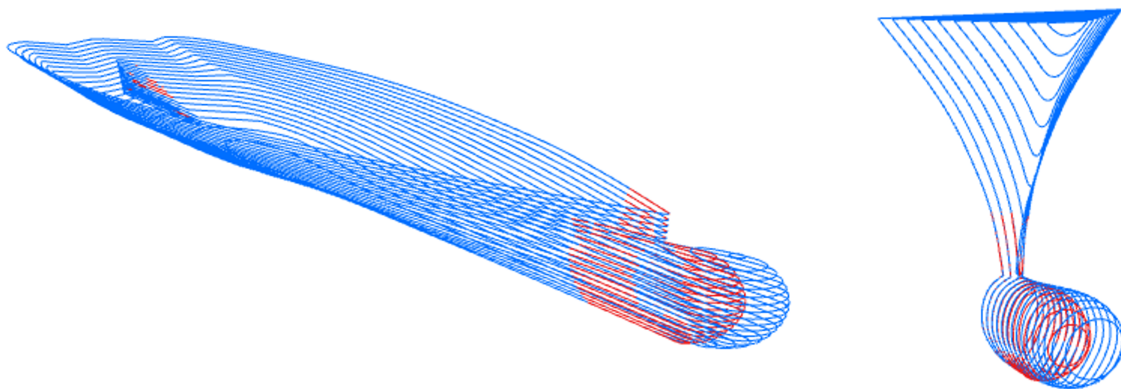


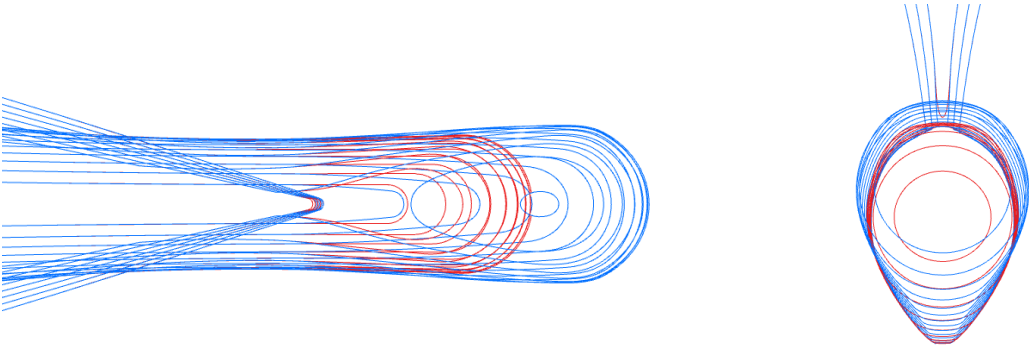
Figure 28: Surrogate Model Comparison for Patrol Vessel

Figure 29 shows a initial and modified hull lines around the bulbous bow region. The modified design and its parameters as evaluated by the optimizer is summarized in Table 12. The reader is further encouraged to refer to Appendix A [7] for additional information pertaining to the use of other surrogate models and hull lines.

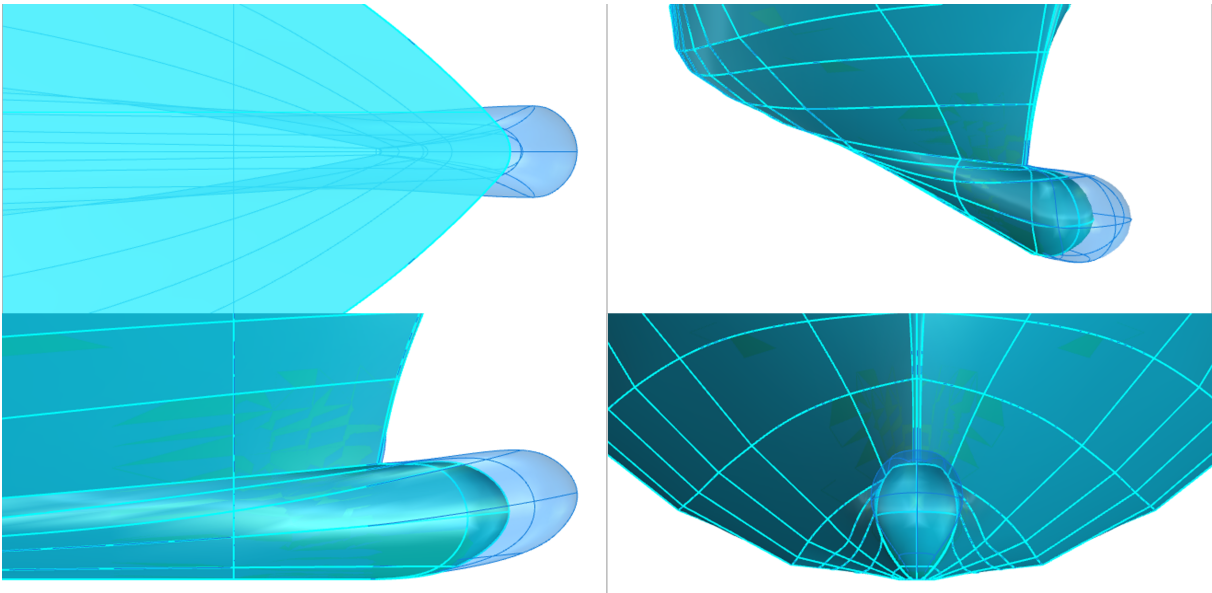
Table 12: Local Optimization results for Patrol Vessel

	L	B	S	A	Change in Resistance
Initial Design	0	1	0	0	0
Design 1 (RBF Model)	1	1.4	0	5.666	-4.26%
Design 2 (KRG Model)	-0.283	1.4	1.5	3.366	-6.55%





**Figure 29:** Bulbous Bow Section Lines (INITIAL & MODIFIED)



**Figure 30:** Bulbous Bow Initial (Shaded) & Modified (Transparent)

## 6 TEST CASE II : CARGO VESSEL

This section presents the optimization workflow of a 82 meter cargo vessel. The primary focus of this test case will be on the global optimization of the hull form and more importantly, to test the Optimization framework for a different category of vessel under slightly modified operating conditions from the previous test case. In addition, another critical aspect of this case was the liberty to showcase and compare the actual resistance values as predicted by the model and estimated by CFD.

### 6.1 Optimization Workflow & Setup

The optimization setup for this vessel was similar to the previous test case with a key modifications to the limits of the design parameters (Ba & Bb) as well as the physical constraints (single bow thruster tunnel). Table 13 and 14 summarizes the vessel characteristics and optimization functions respectively.

#### 6.1.1 Vessel Characteristics

**Table 13:** Characteristics of Cargo Vessel

Name	SXMS-V
Length Overall	82 m
Beam	15.7 m
Height	11 m
Draft	5.5 m
Block Coefficient ( $C_B$ )	0.636
Displacement	4350 tons
Speed	15 knots
Propulsion	Single Screw type
Thruster	1 Bow Thruster
Bulb	'∇' type

### 6.1.2 Objective function & constraints

**Table 14:** Optimization objectives & constraints

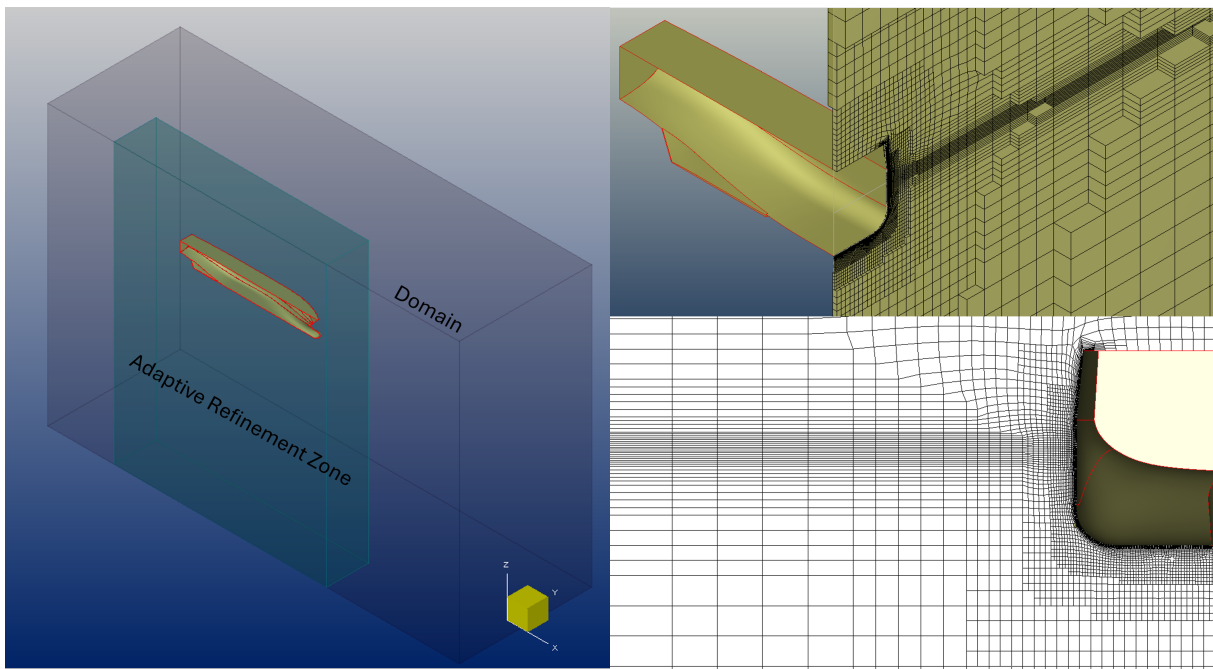
Objective Function	Minimize Surge Resistance at 15 knots
<b>Constraints</b>	
Change in LCB	$\pm 1.50\%$
Change in Displacement	$\pm 1.50\%$
Clashing Objects	1 Bow Thruster Tunnel

### 6.1.3 CFD setup

**Domain Size:** Similar to the earlier test case, the CFD domain was defined as a function of  $L_{pp}$  of the vessel which in-turn was defined in terms of list of coefficients under the 'domain\_coeff' & adaptive\_ref\_coeff parameters in *config.ini* file as shown in Table 5. For the present case, these dimensions are:

Dimension of Outer Domain Zone :  $410 \times 123 \times 164$  m

Dimension of Adaptive Refinement Zone :  $125 \times 125 \times 156$  m



**Figure 31:** Cargo Vessel CFD Domain

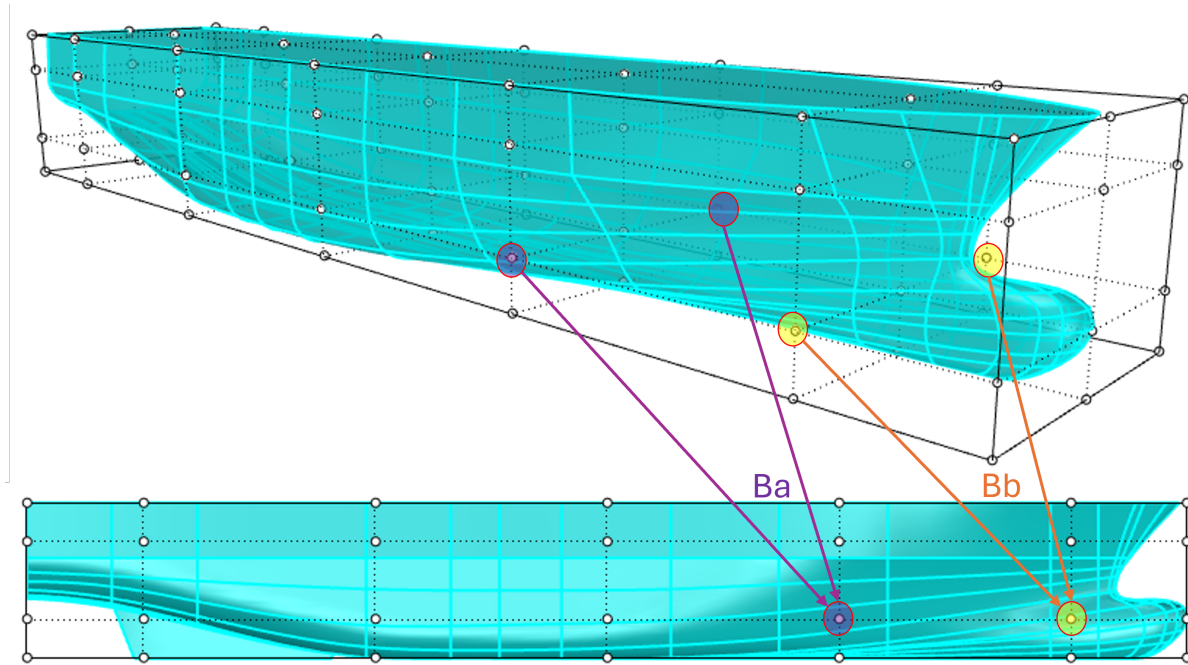
**Table 15:** Cargo Vessel CFD Parameters

Mesh Type	Half Body
Total Cells	$2.504 \times 10^6$
Domain size	410 m (X axis) 123 m (Y axis) 164 m (Z axis)
Global Refinement Diffusion	4
Maximum Number of Refinements	11
Refinement of Wetted Surfaces	7
Refinement of Free Surface	8
Viscous First Layer Thickness ( $Y_{wall}$ )	0.002 m
$Y^+$	205
Viscous Layer Stretching Ratio	1.2
Number of Viscous Layers	17 to 24
Turbulence Model	K-Omega (SST Menter)
Body Motion DOF	Surge (imposed) Sway (fixed) Heave (solved) Roll (fixed) Pitch (solved) Yaw (fixed)
Body Motion Law	1/sinusoidal ramp
External Forces	Drag-based wrench by propeller
Adaptive Grid Refinement	Activated
Number of Time Steps	2000
Time Step Value	0.05 s



## 6.2 Optimization of Fore Part

### 6.2.1 Global cage modifications & Initial Sampling



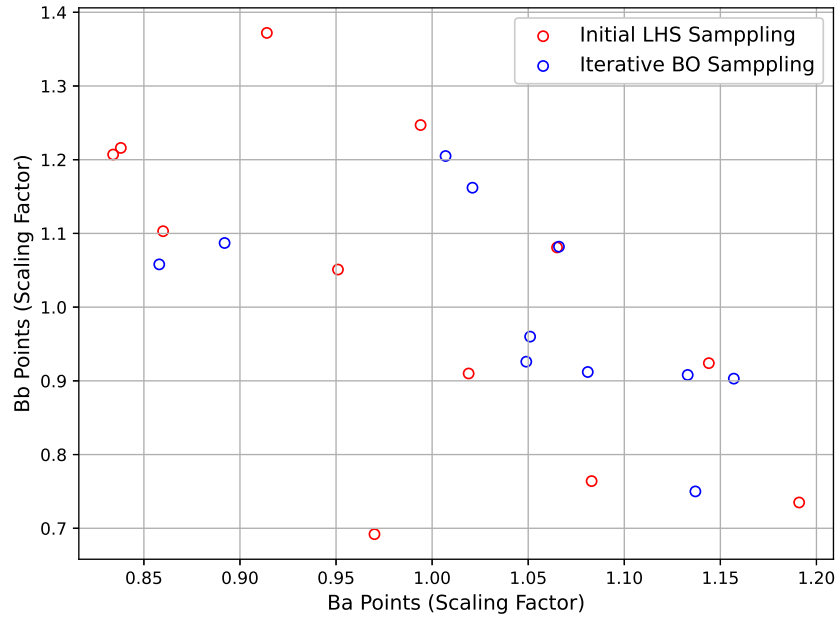
**Figure 32:** Global Design Parameters (Ba,Bb) for Cargo Vessel

The hull model was encapsulated within a cage defined by the parameters mentioned in Table 16. These parameters remain consistent with those selected in the previous test case.

**Table 16:** Global Cage Edit Parameters for Cargo Vessel

Cage Type	Bounding Box
Number of Points along X	7
Number of Points along Y	3
Number of Points along Z	4
Degree of Points in X	2
Degree of Points in Y	2
Degree of Points in Z	2

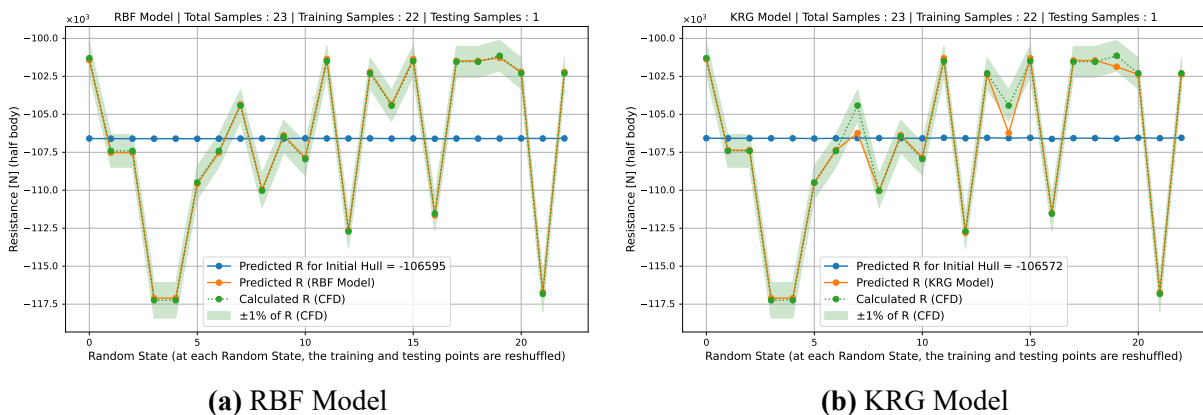
Unlike the previous case, the size of sampling set selected was much smaller (15 samples) to test the predictability of the optimization framework and its ability to perform under a smaller dataset.



**Figure 33:** Design Space Sampling for Global Optimization of Cargo Vessel

In addition to that, 13 out of 15 samples passed the constraint criteria and thus were selected for further CFD analysis. This could be attributed to a much more relaxed constraints and the presence of only a single bow thruster tunnel.

The Python Optimization Framework was then left running undisturbed for the next 50 hours. Moreover, the process encountered some external disruptions such as lost connection to the remote cluster (to which the simulations were being transferred). Although the occurrence of such instances introduced unavoidable errors within the work flow, it also gave the opportunity to fore-see such disruptions and make necessary corrections to the python scripts in terms of additional redundancies and better error handling capacities.



**(a) RBF Model**

**(b) KRG Model**

**Figure 34:** Surrogate Model Comparison for Cargo Vessel

Figure 34 compares the accuracy of two surrogate models i.e. Radial Basis Functions (RBF) & Kriging (KRG) models. here too, it can be observed that the RBF model delivers better results than Kriging model for global optimization where-in the design variables are limited to 2 parameters. An important aspect this case is to consider the accuracy of these models even a limited dataset of sample points. As was mentioned previously, the practice of averaging the results eventually yields a mean value which strongly co-relates with the actual CFD value. This can be demonstrated by comparing the value shown by the 'Predicted Resistance (R) for Initial Hull' in Figure 34 with the actual value as calculated by CFD. This comparison is shown in Table 17.

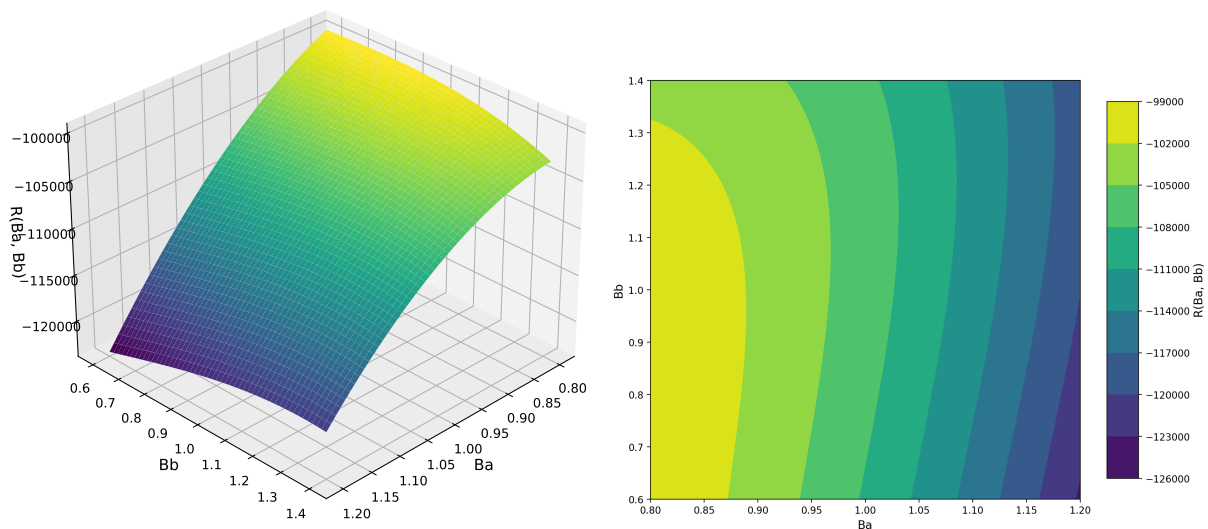
**Table 17:** Initial Resistance Predictions and Deviations

	Resistance (kN)	Deviation from CFD
FineMARINE calculation (CFD)	-106.708	0%
KRG Model Prediction	-106.572	0.1275%
RBF Model Prediction	-106.595	0.1059%

This study was also conducted on various other surrogate models by training them with the same set of data. The reader is encouraged to refer to Appendix B [7] for additional information.

### 6.2.2 Results & Comparison

Figure 35 maps the entire design space which is a function of the two scaling factors (Ba & Bb). RBF model was used as the mapping function to establish the input output relationship between the scaling factors and the corresponding resistance.



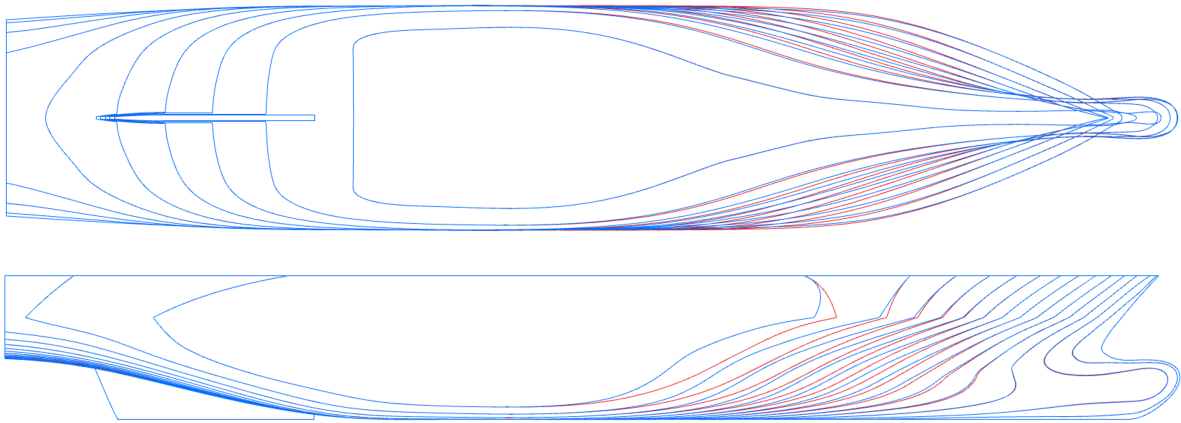
**Figure 35:** Design Space Mapping and Contour for Global Optimization of Cargo Vessel

The maxima of this design space map is evaluated by the optimizer and the results are summarized in Table 18. Design 1, 2 and 3 are based on the converged values of different surrogate models. Design 1 represents the actual CFD value while the resistance values in Design 2 and 3 are predicted by KRG and RBF model respectively. As demonstrated earlier, the model predicted values closely resemble with the actual calculated values for a given combination of design parameters. However, Design 3 can be eliminated on the grounds that the optimizer converged to the limiting values of Ba and Bb which does not fulfill the design constraint. Thus the first 2 designs remains valid and it could be left up to the user to select any of them.

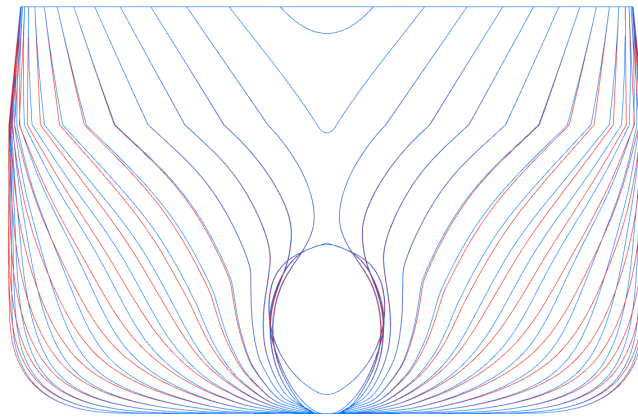
Consequently, design 1 was finalized for further analysis. Figure 36 and Figure 37 draws a comparison of the hull lines between the initial design and final selection.

**Table 18:** Global Optimization results for Cargo Vessel

	Ba	Bb	Resistance (kN)	Change in Resistance (%)
Initial Design	1	1	-106.708	0
Design 1	0.858	1.058	-101.143	-5.22%
Design 2	0.802	1.193	-99.671	-6.59%
Design 3	0.800	0.800	-99.486	-6.77%



**Figure 36:** Cargo Vessel Hull Lines (**INITIAL** & **MODIFIED**)



**Figure 37:** Cargo Vessel Section Lines (**INITIAL** & **MODIFIED**)

One can also observe the wave elevation plot in Figure 38 which shows a slightly reduced amplitude of wave elevation for optimized design as against the initial design. It must be noted that the elevation is measured from the keel of the vessel for an operational draft of 5.5 meters at 15 knots.

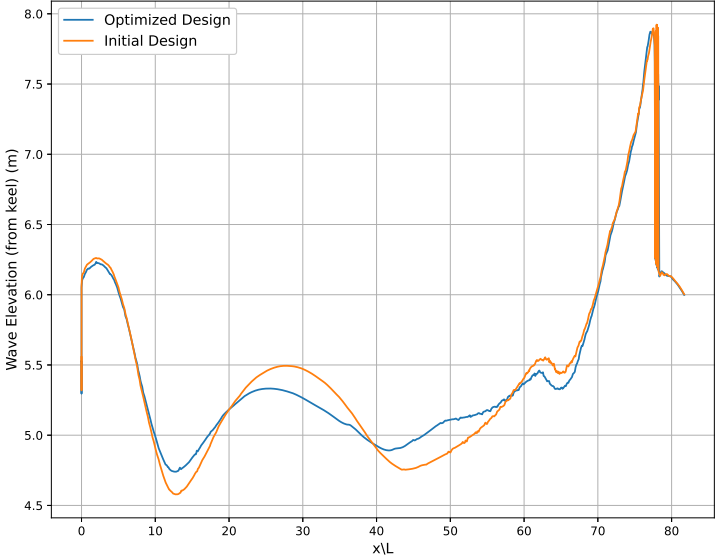


Figure 38: Cargo Vessel Global Wave Elevation Comparison

## 7 REMARKS & CONCLUSION

A comprehensive surrogate model based optimization framework was developed and tested on two different hull forms with promising results and satisfactory performance. While the framework was built from scratch, it can lay foundation for future developments and allows us to reframe conventional optimization strategies with modern tools and methods. In this regard, brief remarks on some of the advantages and limitations of the developed framework as well as the future scope of work is presented in this section.

### Advantages

**Robust Framework:** The developed framework demonstrates a high degree of robustness by being capable of accepting and processing various hull forms. This versatility makes it suitable for a wide range of design applications.

**User Flexibility:** The framework provides users with flexibility to govern shape modifications by imposing their own defined limits. This enables customized design adjustments, making the optimization process more aligned with specific user requirements.

**Incorporation of Constraints:** The framework is designed to consider both physical constraints (such as the positions of tanks, gearboxes, and thrusters) as well as user-defined hydrostatic constraints (such as changes in metacentric height or hull form coefficients). This ensures that the optimization process remains realistic while also adhering to essential design parameters.

**Minimal User Intervention:** Minimizing the need for user interference reduces the sensitivity of the workflow to external influences. This streamlined process enhances the reliability of the optimization results.

**Reduction in Computational Load:** The integration of surrogate models within the framework allows for a significant reduction in the number of computationally expensive simulations. This approach distinguishes the developed framework from conventional optimization practices which are often computationally demanding in nature.

### Limitations

**Dependency on External Tools:** The framework is composed of multiple independent tools and applications, which makes it vulnerable to any significant changes within these components. Thus the users must be well-acquainted with the framework's functioning, to troubleshoot any potential issues that may arise from tool dependencies.

**Requirement for Intuitive Understanding:** The initial setup of the optimization requires users to have some level of intuitive understanding of free-form deformation, particularly for the selection of cage points and definition of deformation vectors. This could pose a limiting factor for wider acceptability of the framework amongst users.

**Sensitivity of Optimizer:** The optimizer within the framework exhibits sensitivity to its underlying parameters, such as the choice of acquisition function, infill criteria, and the general issue of the "curse of dimensionality". These factors can influence the optimization results and thus require careful consideration and fine-tuning by the user.

### Future Scope

**Comprehensive Sensitivity Analysis:** Future work could involve conducting a detailed sensitivity analysis of various surrogate models and their underlying parameters. Additionally, developing an ensemble of multiple surrogates could help mitigate potential biases introduced by individual models, leading to more reliable optimization results.

**Study of Design Space:** The model can be utilized to perform a comprehensive study of the design space by examining the sensitivity of individual design variable on the overall performance metrics. This can further help in evaluating the merits and potential of various design variables.

**Multi-Objective Optimization:** Expanding the framework to handle multiple objective functions could greatly enhance its utility. For example, optimizing for seakeeping characteristics alongside propulsion efficiency would provide a



more holistic approach to vessel design, taking into account both performance and operational efficiency.

**Optimization for Multiple Speeds:** Further development could involve adding features to optimize for multiple operating speeds simultaneously. By defining a single objective function as a weighted average of these speeds, the framework could provide more comprehensive and versatile design solution that is optimized across a range of loading conditions.

---

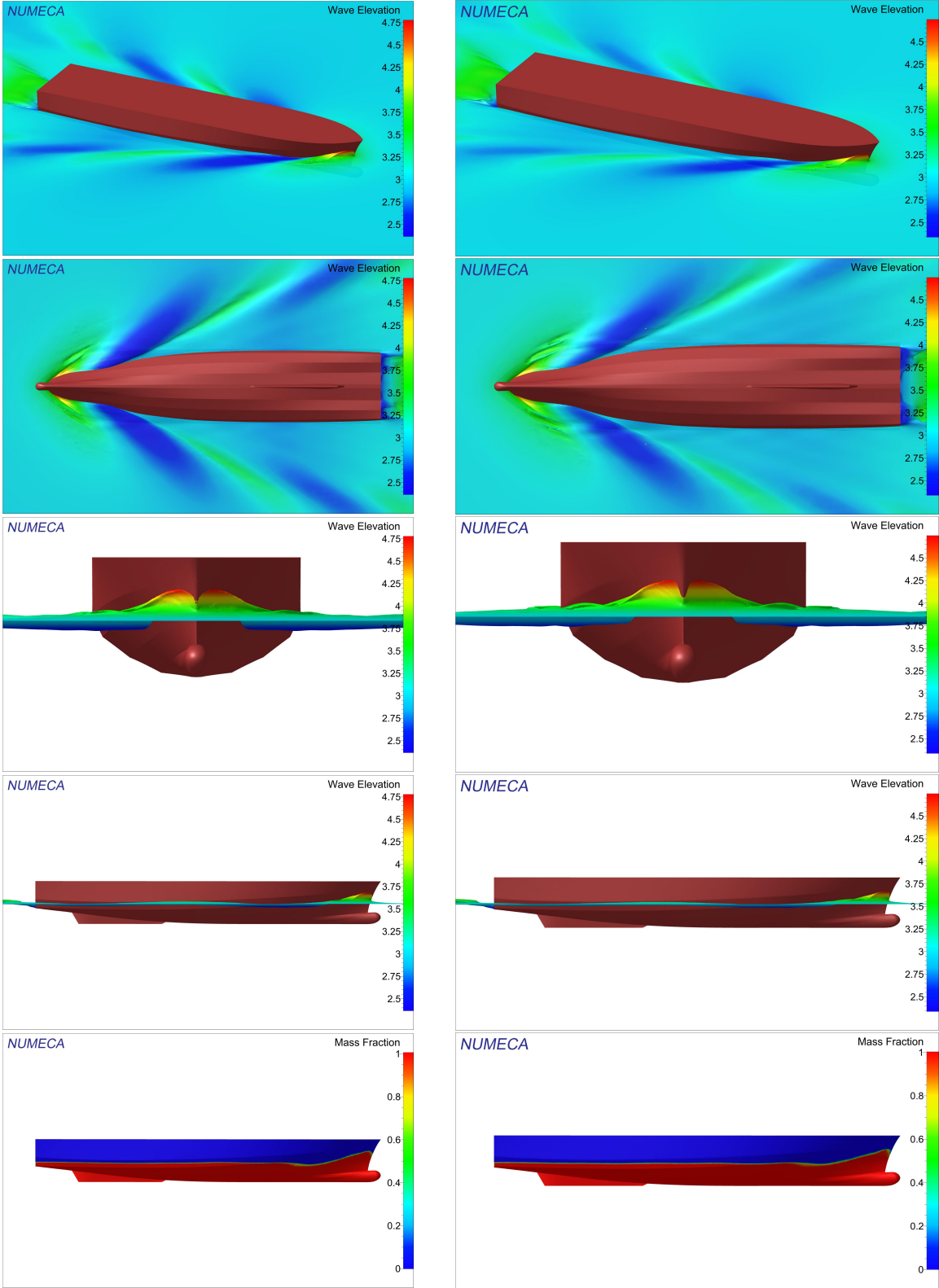
**REFERENCES**

- [1] Y. Feng, O. el Moctar, and T. E. Schellin, “Parametric hull form optimization of containerships for minimum resistance in calm water and in waves,” *Journal of Marine Science and Application*, vol. 20, pp. 670–693, 4 Dec. 2021, ISSN: 19935048. DOI: 10.1007/s11804-021-00243-w.
- [2] S. Han, Y. S. Lee, and Y. B. Choi, *Hydrodynamic hull form optimization using parametric models*, Mar. 2012. DOI: 10.1007/s00773-011-0148-8.
- [3] N. R. Council, *Twenty-First Symposium on Naval Hydrodynamics*. Washington, DC: The National Academies Press, 1997, ISBN: 978-0-309-05879-7. DOI: 10.17226/5870. [Online]. Available: <https://nap.nationalacademies.org/catalog/5870/twenty-first-symposium-on-naval-hydrodynamics>.
- [4] J. Čerka, R. Mickevičienė, Ž. Ašmontas, L. Norkevičius, T. Žapnickas, V. Djačkov, and P. Zhou, “Optimization of the research vessel hull form by using numerical simulation,” *Ocean Engineering*, vol. 139, pp. 33–38, 2017, ISSN: 0029-8018. DOI: <https://doi.org/10.1016/j.oceaneng.2017.04.040>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0029801817302238>.
- [5] A. Kracht, “Design of bulbous bows,” *SNAME Transactions*, vol. 86, 1978.
- [6] X. Liu, W. Zhao, and D. Wan, “Multi-fidelity co-kriging surrogate model for ship hull form optimization,” *Ocean Engineering*, vol. 243, p. 110 239, 2022, ISSN: 0029-8018. DOI: <https://doi.org/10.1016/j.oceaneng.2021.110239>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0029801821015523>.
- [7] P. Wang, Y. Feng, Z. Chen, and Y. Dai, “Study of a hull form optimization system based on a gaussian process regression algorithm and an adaptive sampling strategy, part i: Single-objective optimization,” *Ocean Engineering*, vol. 279, p. 114 502, 2023, ISSN: 0029-8018. DOI: <https://doi.org/10.1016/j.oceaneng.2023.114502>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0029801823008867>.
- [8] K. Jakub and M. Radomil, “Recent advances and applications of surrogate models for finite element method computations: A review,” *Soft Computing*, vol. 26, pp. 13 709–13 733, 24 Dec. 2022, ISSN: 14337479. DOI: 10.1007/s00500-022-07362-8.
- [9] A. Forrester, A. Sobester, and A. Keane, *Engineering design via surrogate modelling : a practical guide*. Wiley, 2008, p. 210, ISBN: 9780470060681.
- [10] P. Jiang, Q. Zhou, and X. Shao, *Surrogate Model-Based Engineering Design and Optimization*. Jan. 2020, ISBN: 978-981-15-0730-4. DOI: 10.1007/978-981-15-0731-1.

- 
- [11] P. Saves, R. Lafage, N. Bartoli, Y. Diouane, J. Bussemaker, T. Lefebvre, J. T. Hwang, J. Morlier, and J. R. R. A. Martins, “SMT 2.0: A surrogate modeling toolbox with a focus on hierarchical and mixed variables gaussian processes,” *Advances in Engineering Software*, vol. 188, p. 103 571, 2024. DOI: <https://doi.org/10.1016/j.advengsoft.2023.103571>.
- [12] F. Nogueira, *Bayesian Optimization: Open source constrained global optimization tool for Python*, 2014–. [Online]. Available: <https://github.com/bayesian-optimization/BayesianOptimization>.
- [13] J. Helton and F. Davis, “Latin hypercube sampling and the propagation of uncertainty in analyses of complex systems,” *Reliability Engineering & System Safety*, vol. 81, no. 1, pp. 23–69, 2003, ISSN: 0951-8320. DOI: [https://doi.org/10.1016/S0951-8320\(03\)00058-9](https://doi.org/10.1016/S0951-8320(03)00058-9). [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0951832003000589>.
- [14] R. Jin, W. Chen, and A. Sudjianto, “An efficient algorithm for constructing optimal design of computer experiments,” *Journal of Statistical Planning and Inference*, vol. 134, pp. 268–287, 1 Sep. 2005, ISSN: 0378-3758. DOI: 10.1016/J.JSPI.2004.02.014.
- [15] J. R. R. A. Martins and S. A. (A. Ning, *Engineering design optimization*. 2021, p. 637, ISBN: 9781108833417.
- [16] T. W. Sederberg and S. R. Parry, “Free-form deformation of solid geometric models.,” *Computer Graphics (ACM)*, vol. 20, 4 1986, ISSN: 00978930. DOI: 10.1145/15886.15903.
- [17] J. Reid, *Free-form deformation (ffd)*, 2021. [Online]. Available: <https://github.com/mdolab/pygeo>.
- [18] J. A. Samareh, *Aerodynamic shape optimization based on free-form deformation*. [Online]. Available: <http://mdob.larc.nasa.gov>, .
- [19] *FINE/Marine users guide*, version 10.2, CADENCE DESIGN SYSTEMS, 2024.
- [20] J. S. Gray, J. T. Hwang, J. R. R. A. Martins, K. T. Moore, and B. A. Naylor, “OpenMDAO: An open-source framework for multidisciplinary design, analysis, and optimization,” *Structural and Multidisciplinary Optimization*, vol. 59, no. 4, pp. 1075–1104, Apr. 2019. DOI: 10.1007/s00158-019-02211-z.
- [21] Robert McNeel & Associates, *Rhino user guides*, Robert McNeel & Associates, 2024. [Online]. Available: <https://developer.rhino3d.com/guides/>.

# APPENDIX A : TEST CASE I (GENERIC PATROL VESSEL)

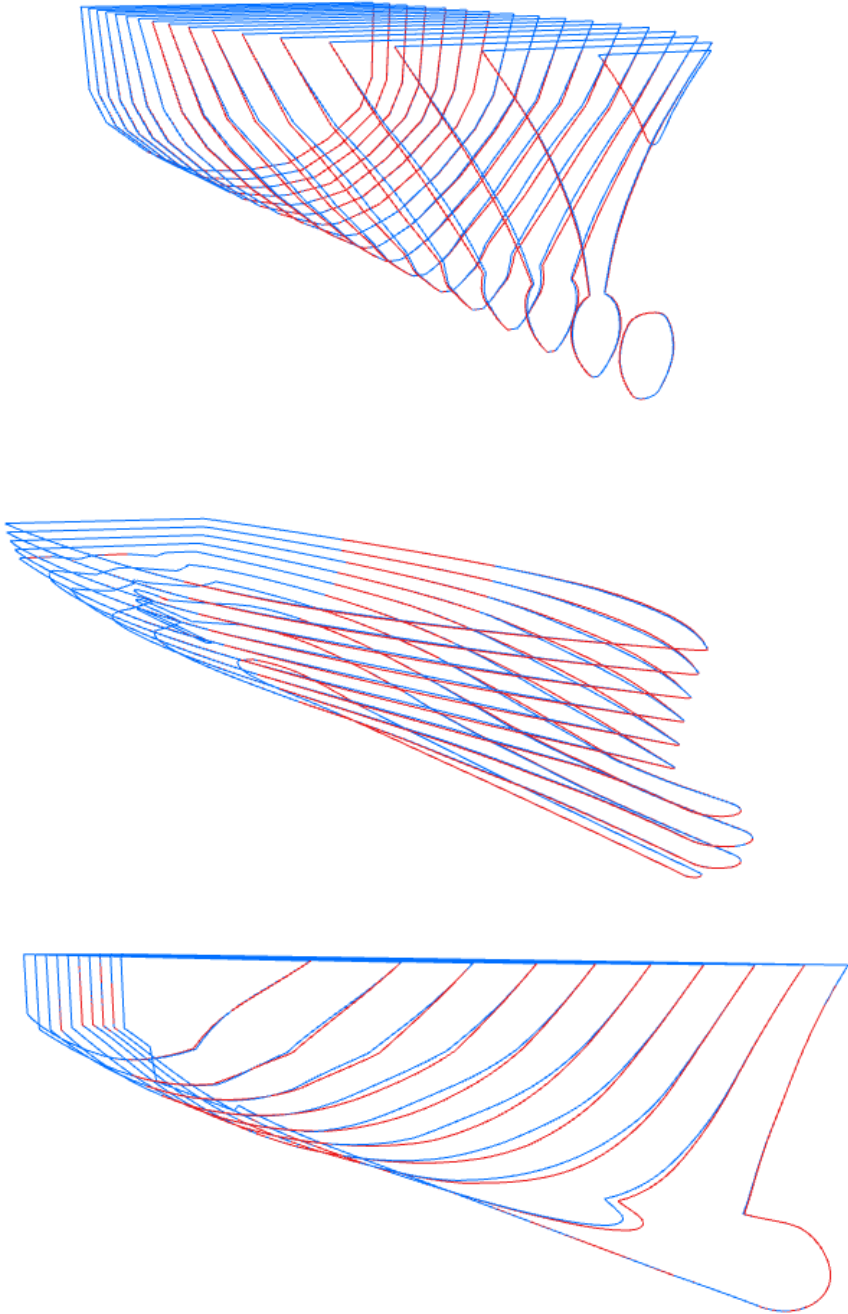
## Global Optimization of Fore part of a Patrol Vessel



Test Case I (Fore Part Modification): Initial Design (left) & Modified Design (right)

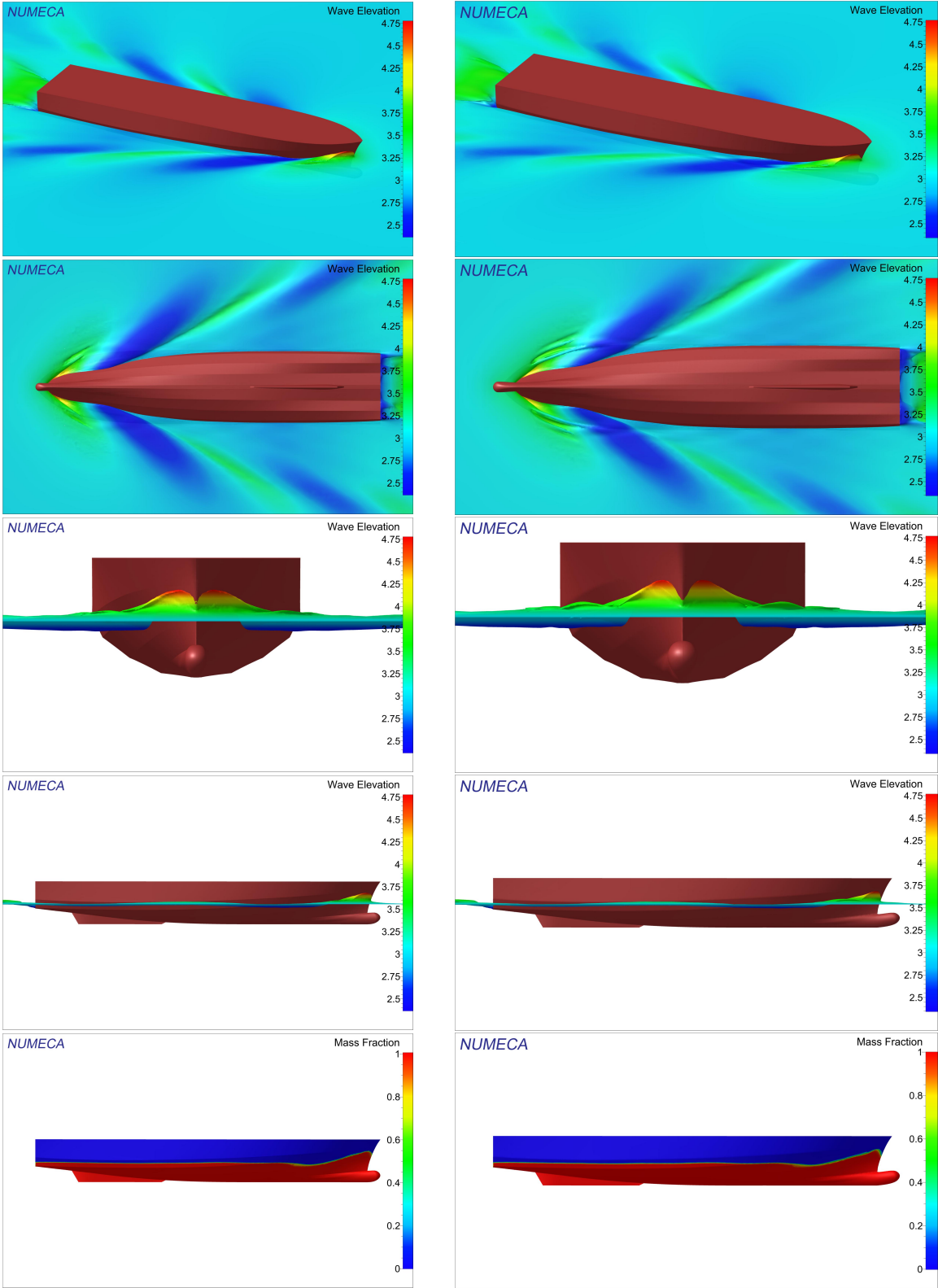


Patrol Vessel Hull Lines (INITIAL & MODIFIED)

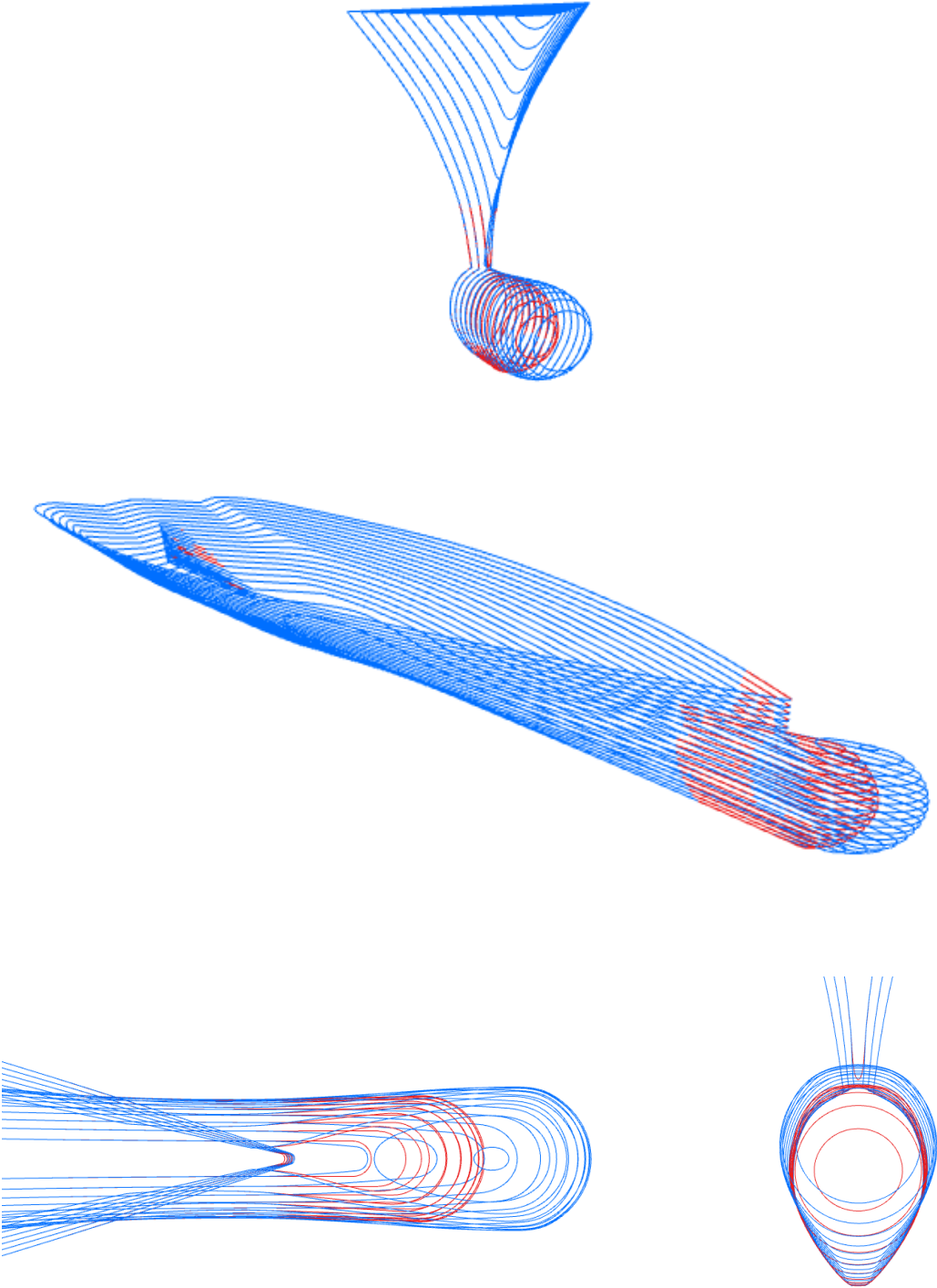


Patrol Vessel Hull Lines (INITIAL & MODIFIED)

Local Optimization of Bulbous Bow of a Patrol Vessel



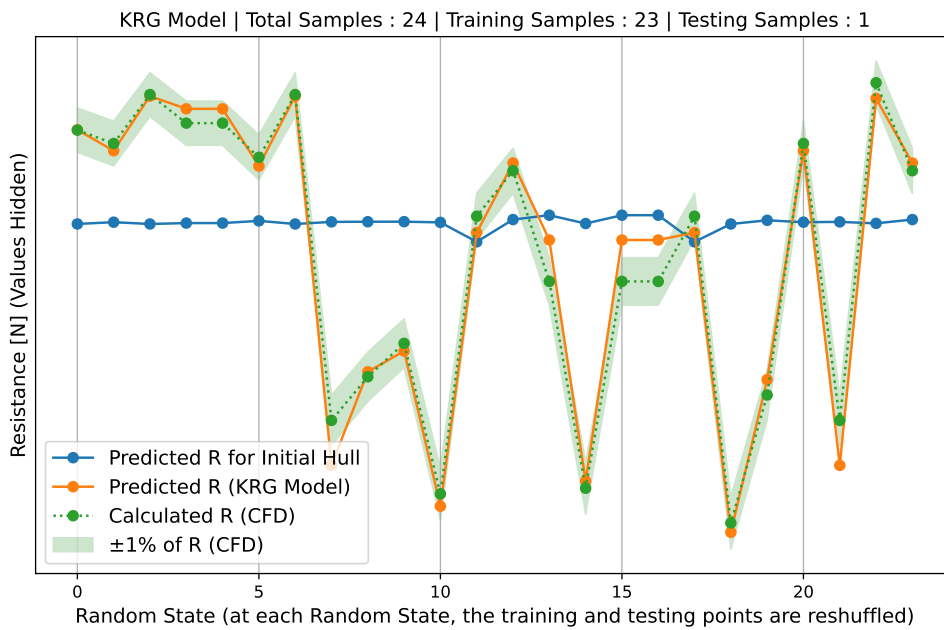
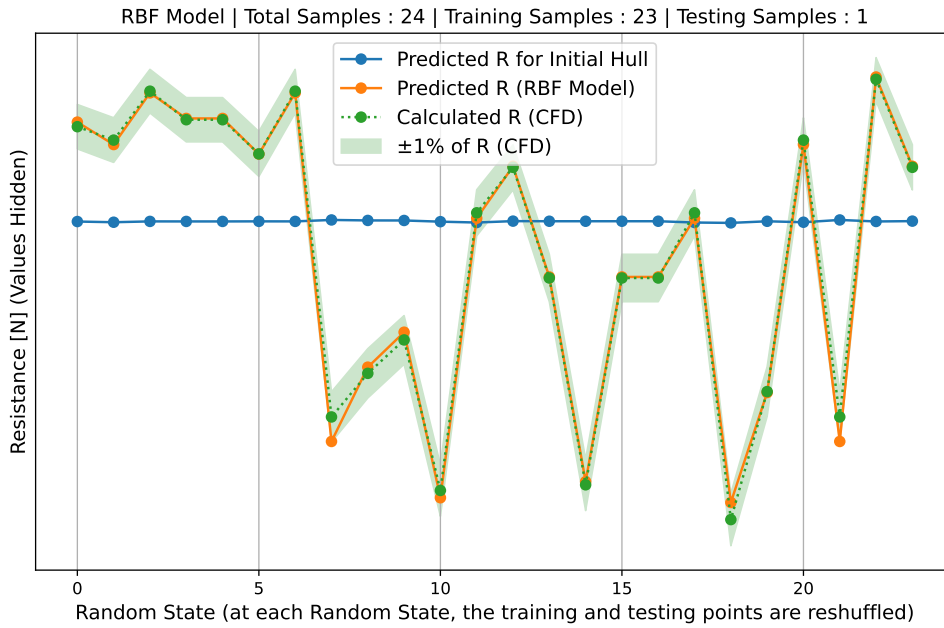
Test Case I (Bulb Modification): Initial Design (left) & Modified Design (right)



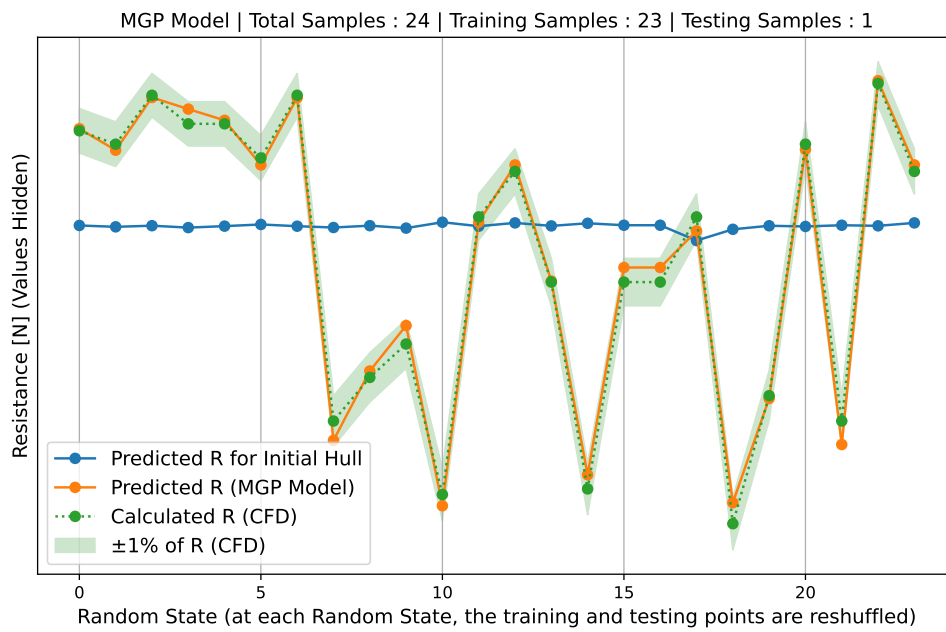
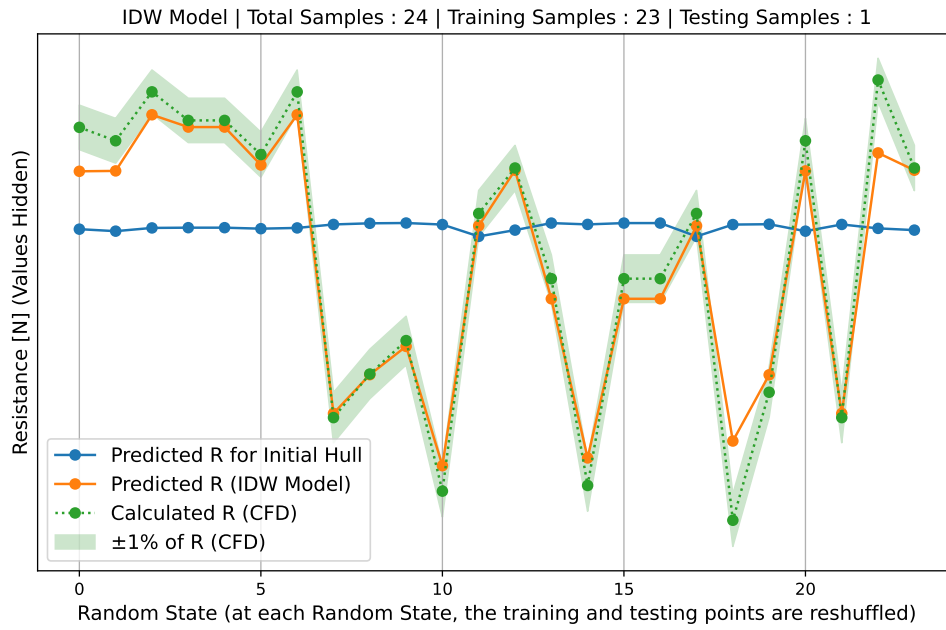
Patrol Vessel Hull Lines (INITIAL & MODIFIED)



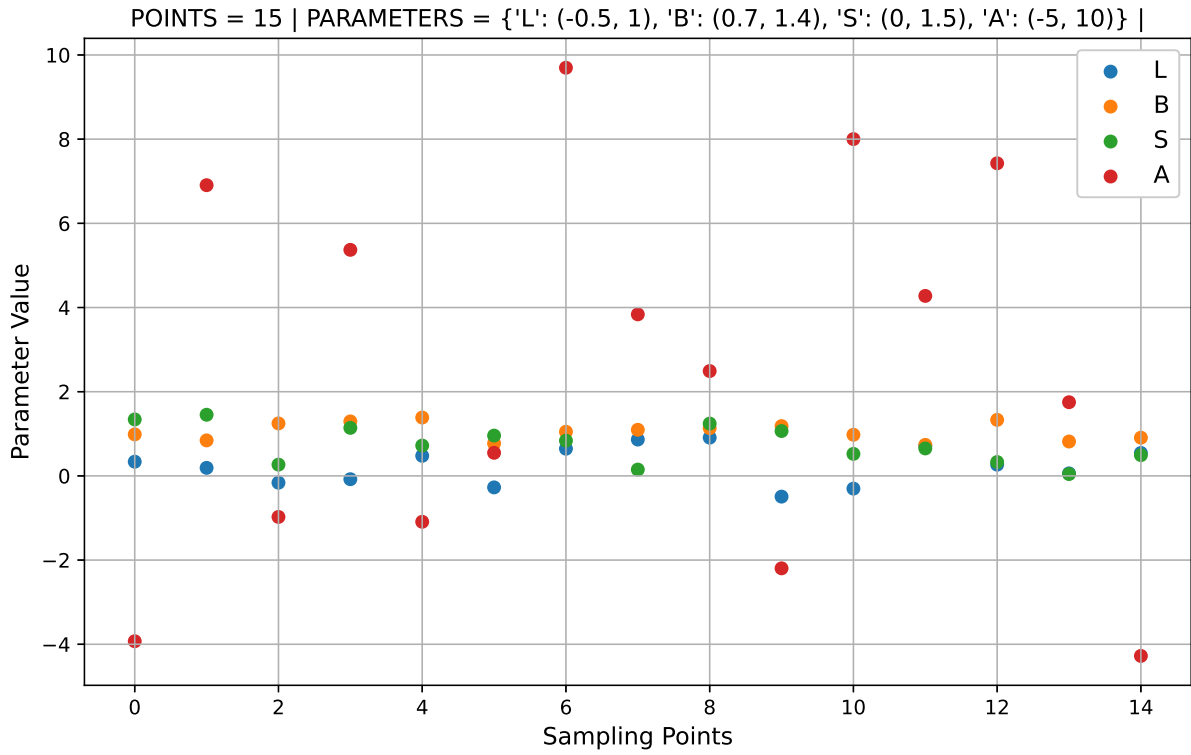
Surrogate Model Results for Patrol Vessel



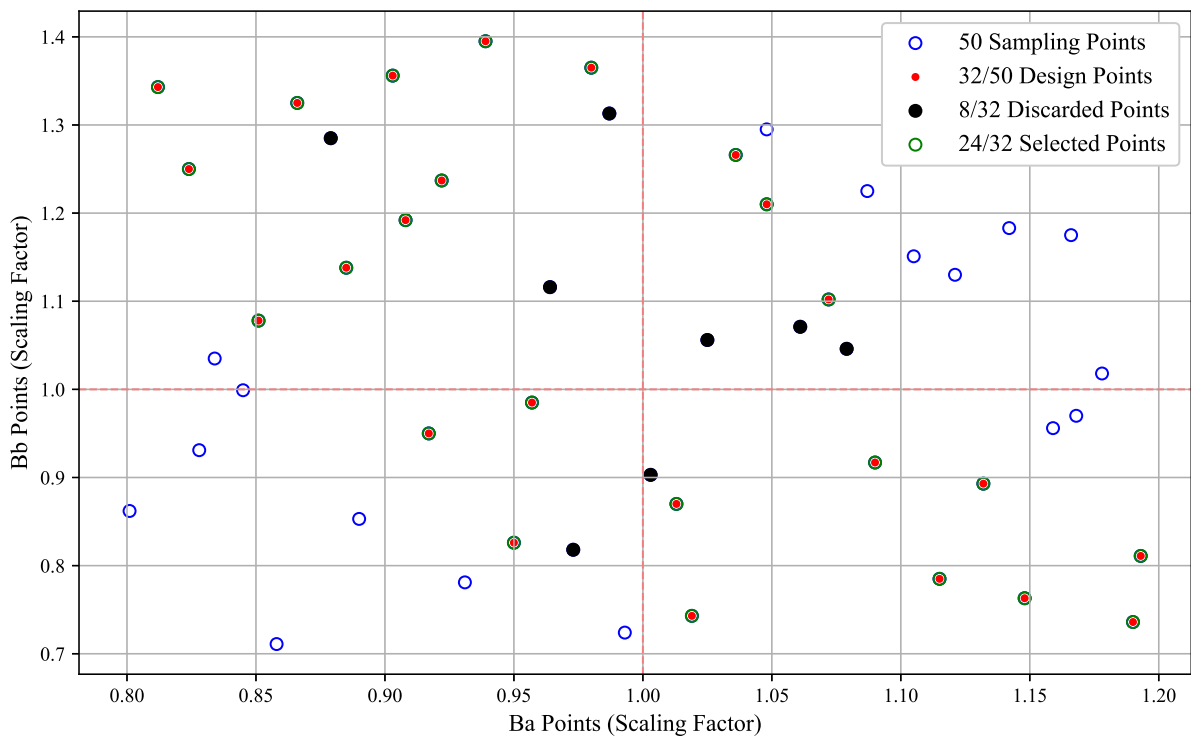
Surrogate Models (RBF & KRG) Comparison for Patrol Vessel



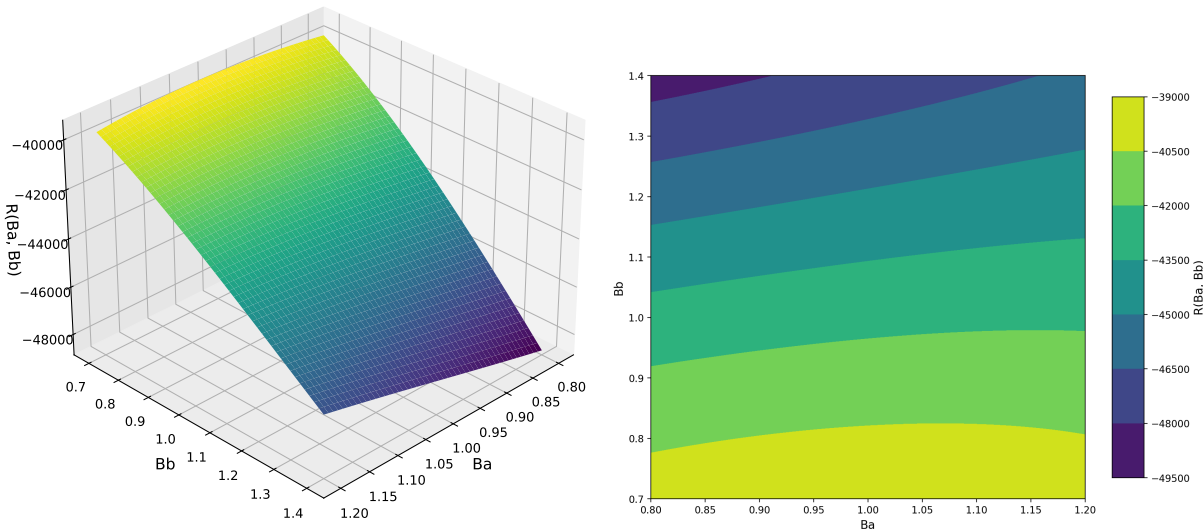
Surrogate Models (IWD & MGP) Comparison for Patrol Vessel



Design Space Sampling for Local Optimization of Patrol Vessel



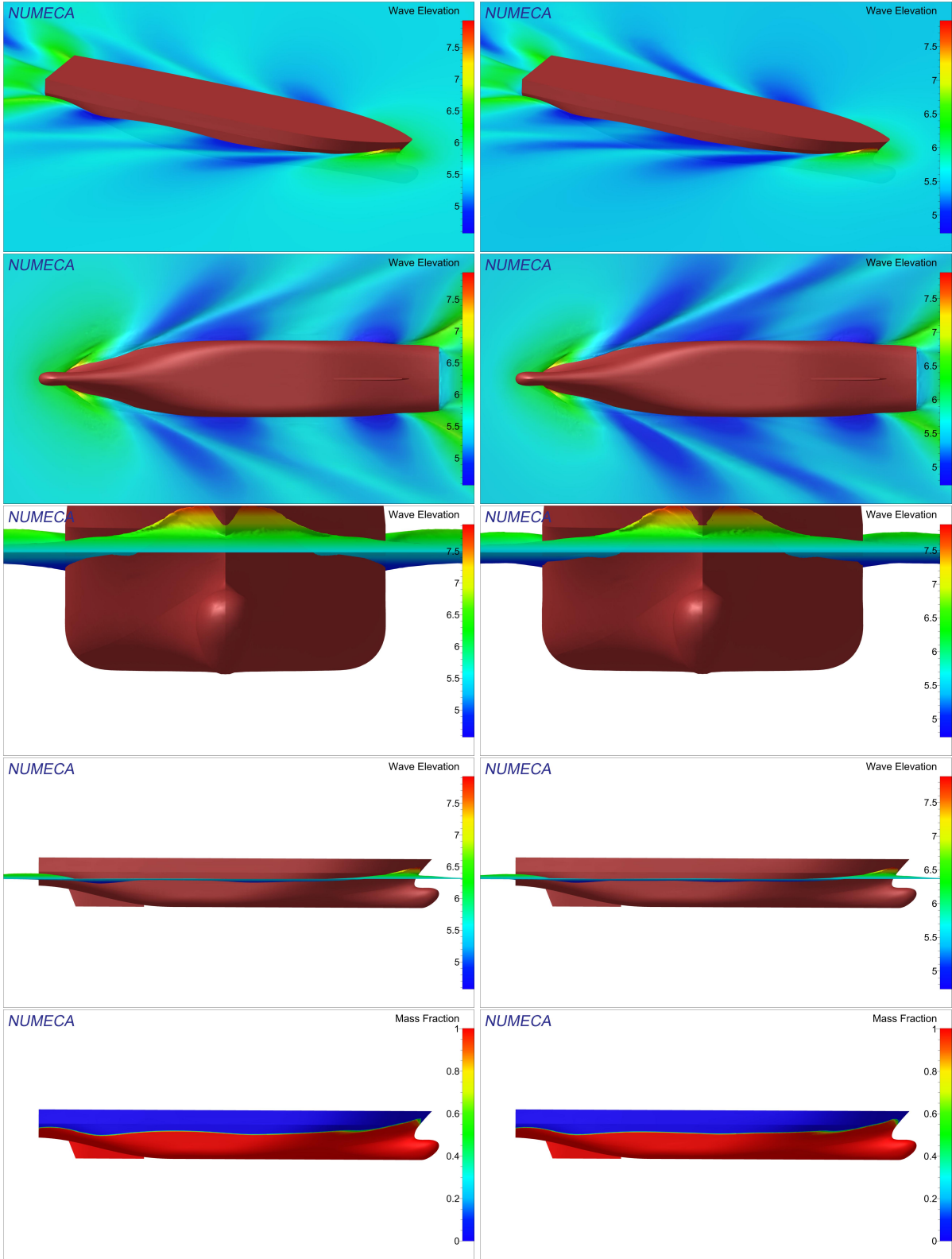
Design Space Sampling for Global Optimization of Patrol Vessel



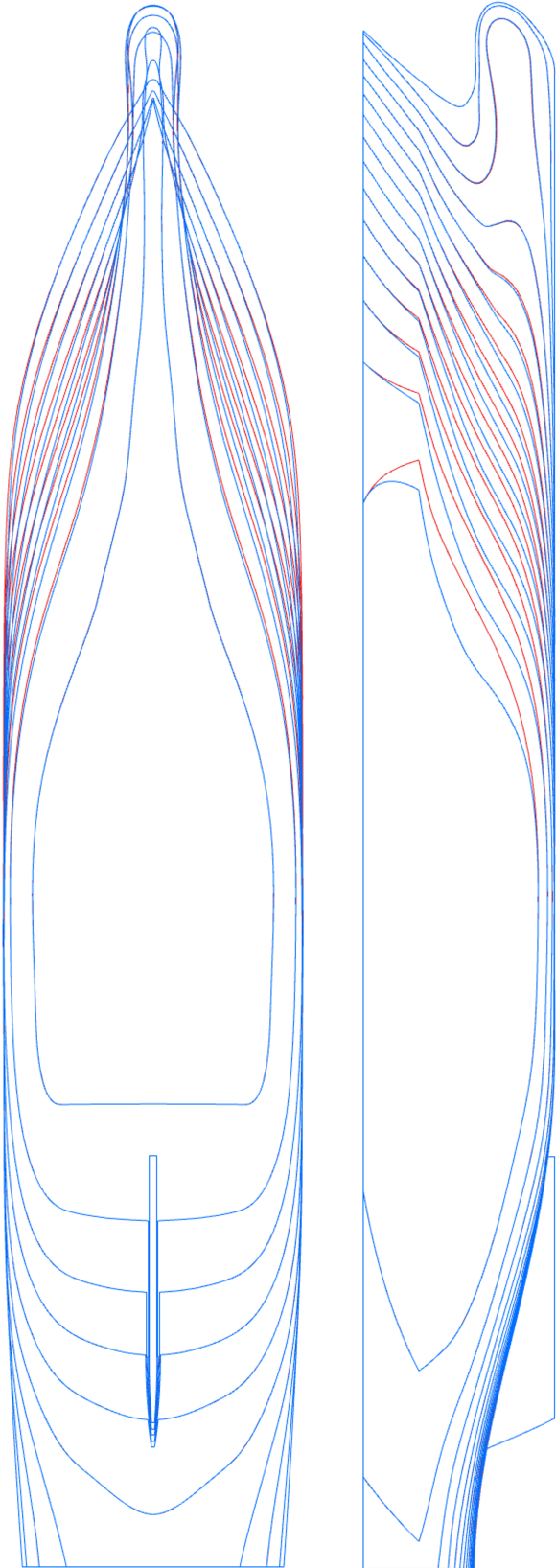
Design Space Mapping and Contour for Global Optimization of Patrol Vessel

# APPENDIX B : TEST CASE II (SXMSV)

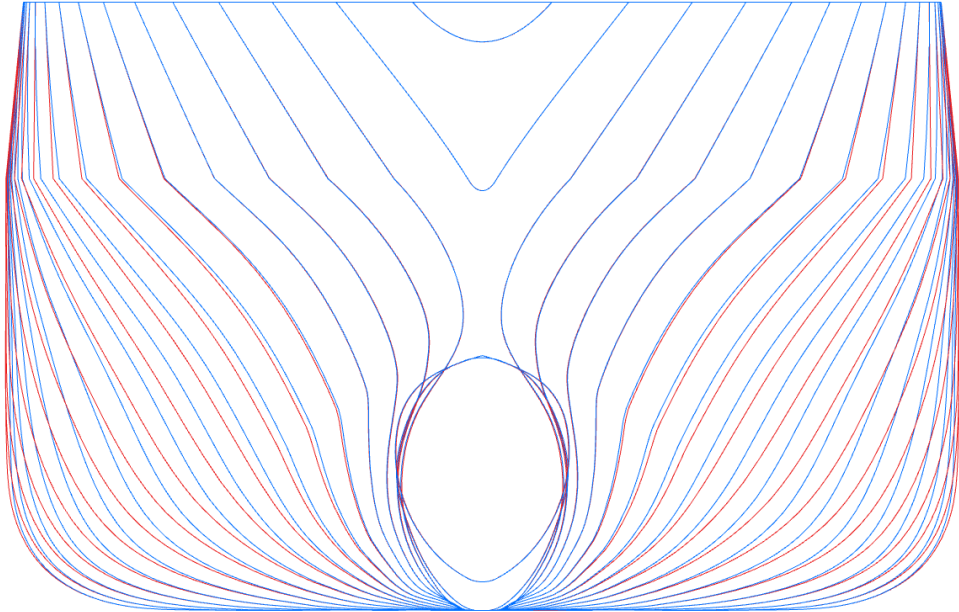
## Global Optimization of Fore part of SXMSV



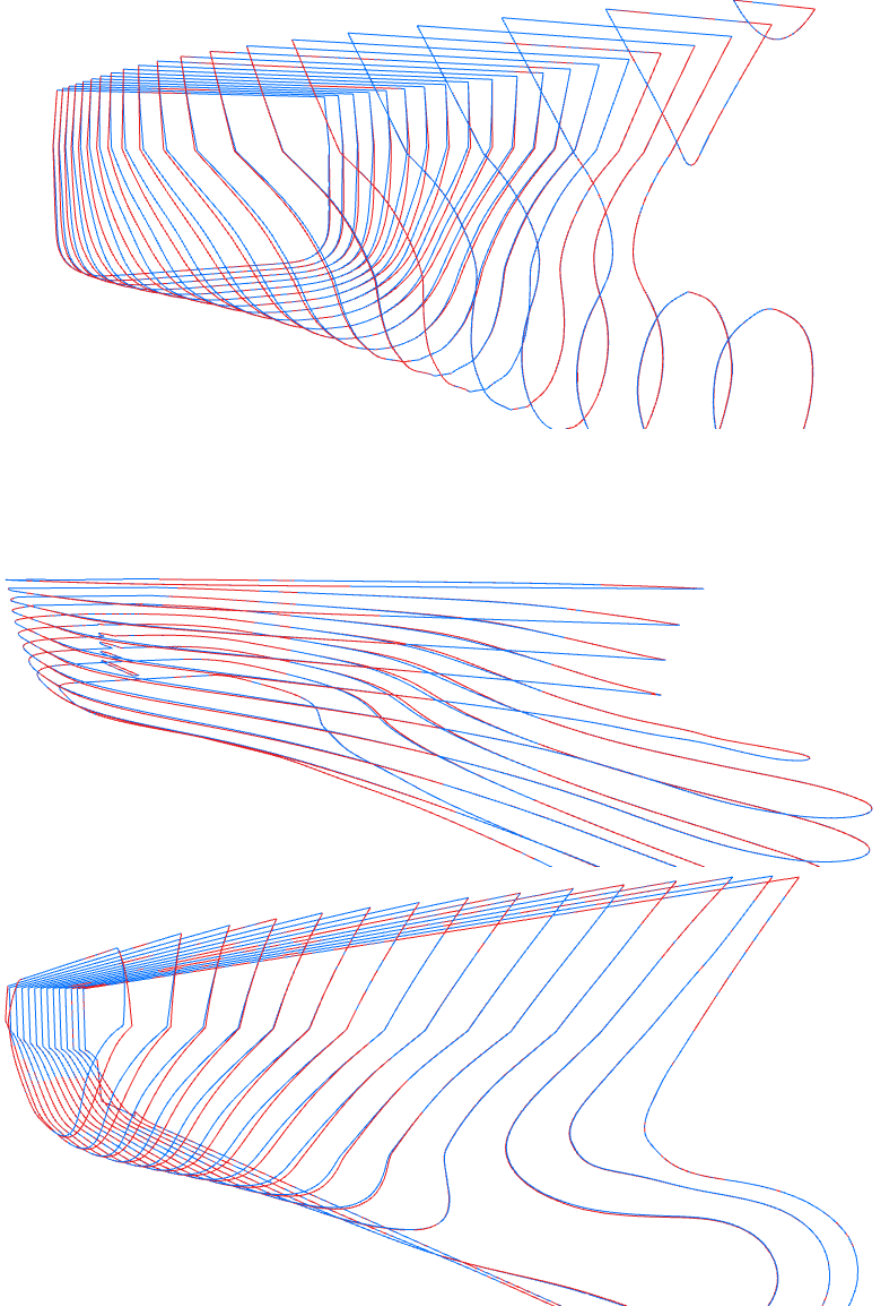
Test Case II (Fore Part Modification): Initial Design (left) & Modified Design (right)



SXMSV Hull Lines (INITIAL & MODIFIED)



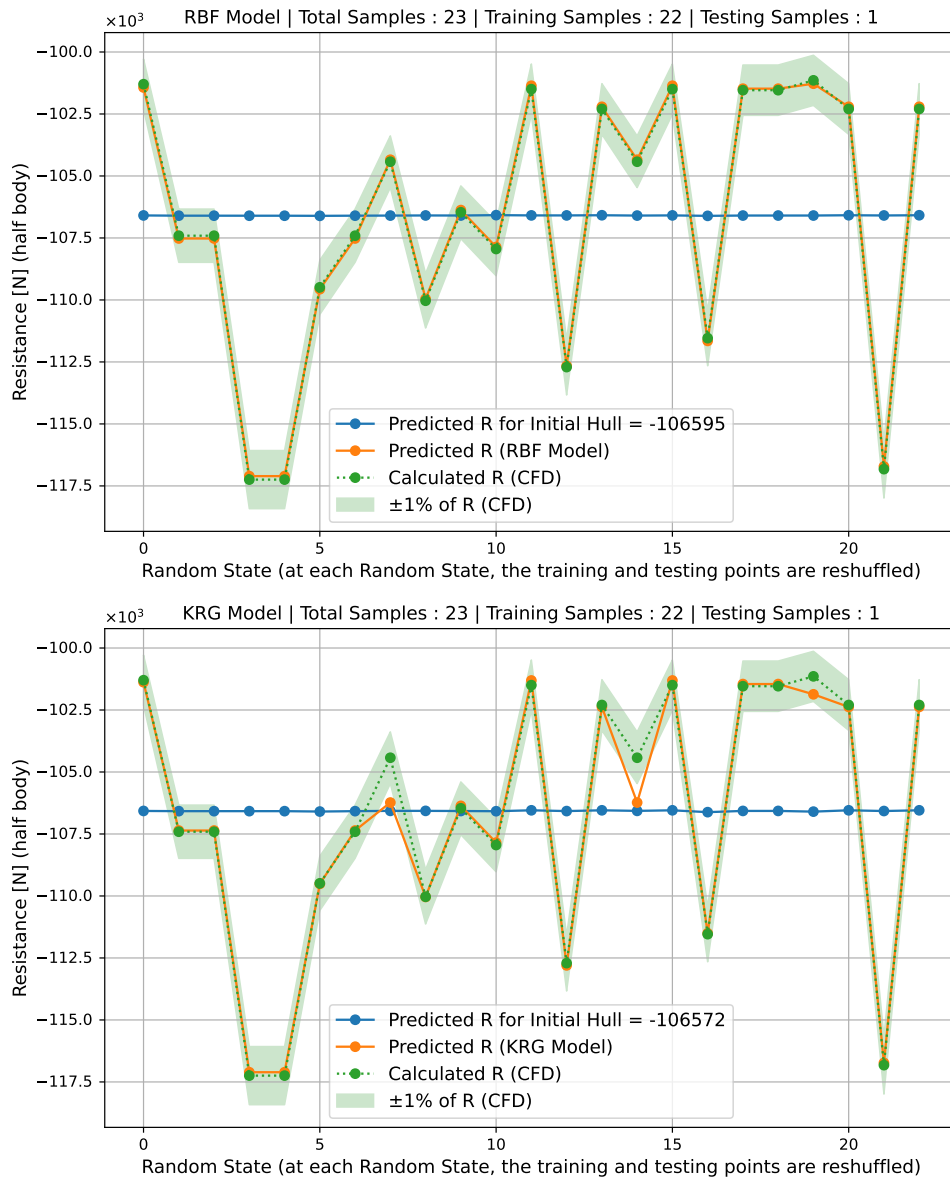
SXMSV Section Lines (INITIAL & MODIFIED)



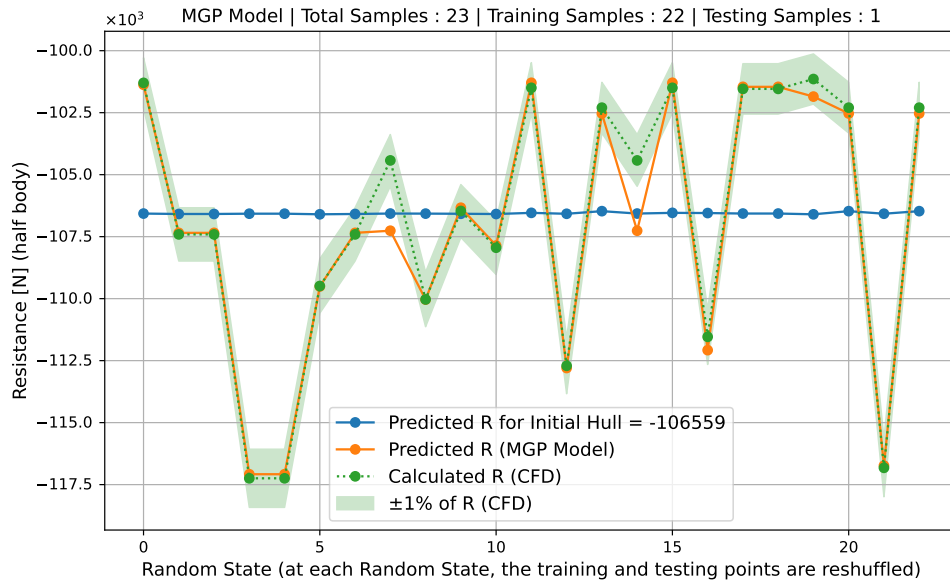
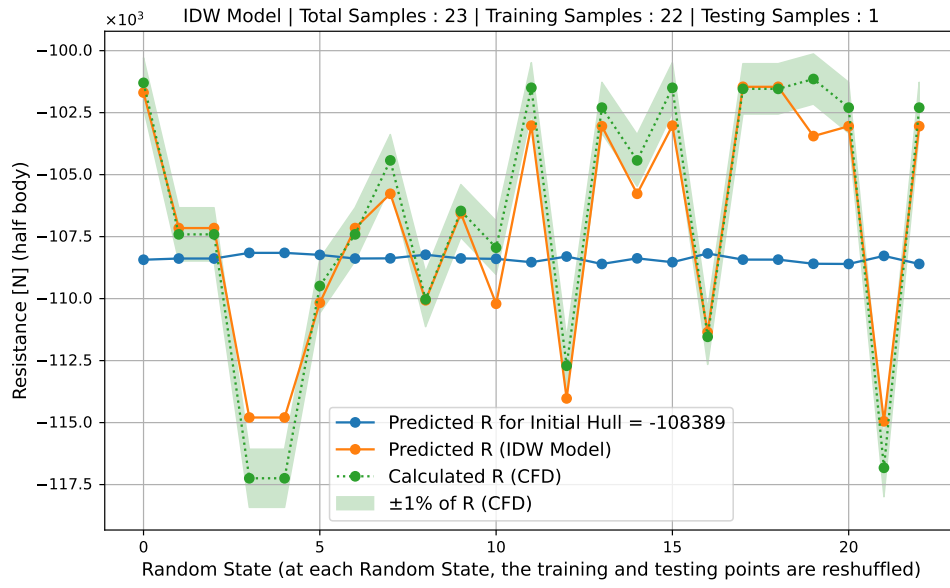
SXMSV Hull Lines (INITIAL & MODIFIED)



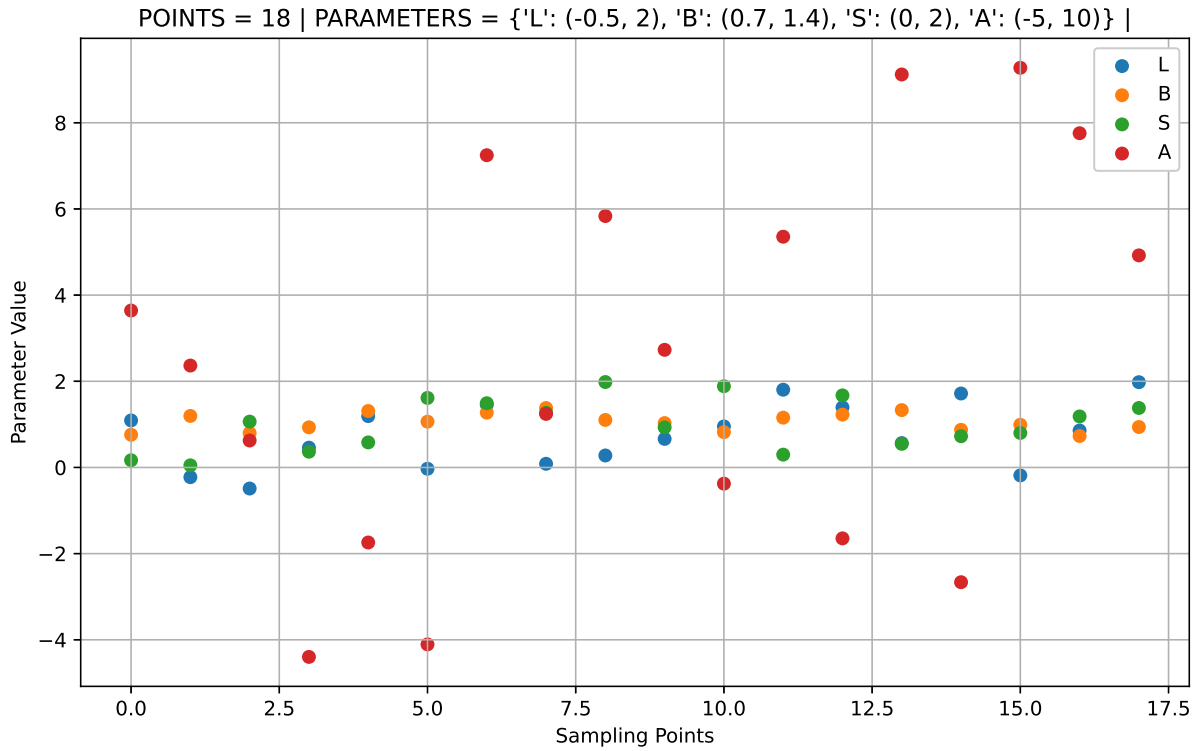
Surrogate Model Results for SXMSV



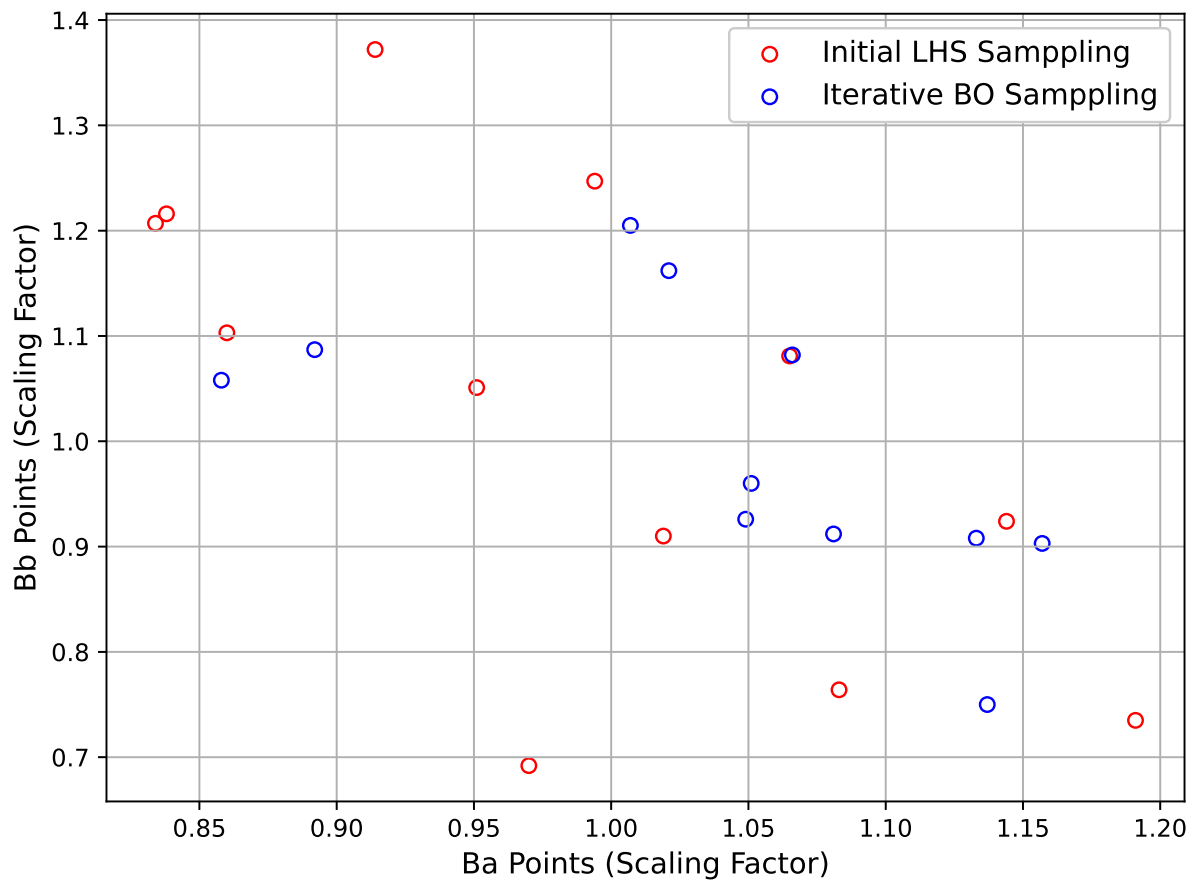
Surrogate Models (RBF & KRG) Comparison for SXMSV



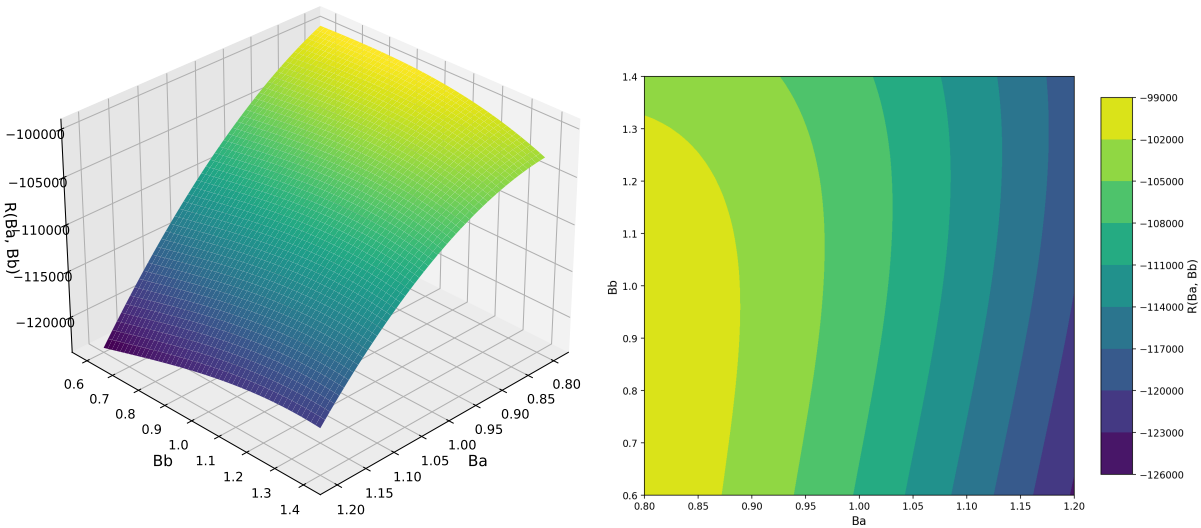
Surrogate Models (IWD & MGP) Comparison for SXMSV



Design Space Sampling for Local Optimization of SXMSV



Design Space Sampling for Global Optimization of SXMSV



Design Space Mapping and Contour for Global Optimization of SXMSV