Surrogate based optimisation for hull shapes with DAKOTA toolkit

Thesis for the degree of MSc in Marine Technology in the specialisation of Ship Hydromechanics

Surrogate based optimisation for hull shapes with DAKOTA toolkit

By
Silvia Giorgia Casella

Performed at
Van Oossanen Naval Architects

June 30, 2020

This thesis is classified as confidential in accordance with the general conditions for projects performed by the TUDelft.

Company
Daily Supervisor: Vincent Jacobs

Thesis exam committee
Chair/Responsible Professor: Jerry Westerweel
Staff Member: Ido Akkerman
Staff Member: Robert Hekkenberg
Company Member: Vincent Jacobs

Author Details
Studynumber: 4897633

"Evviva le navi!
Con il loro ansimare, scuotere, sospirare; con il loro gioire delle carezze delle onde, con il loro godere nell'amplesso del mare, le navi sono a misura d'uomo. Teniamole in vita come una prova d'amore. Usiamole per far felici gli ultimi romantici."

"Long live the ships!
With their panting, shaking, sighing; with their joy of the caresses of the waves, with their pleasure in the embrace of the sea, the ships are on a human scale. Let's keep them alive as a test of love. Let's use them to make the last romantics happy."

– Tiziano Terzani

Summary

The thesis proposes a methodology for a Surrogate Based Optimisation (SBO) study and its application in the Naval Architecture field using the open-source software DAKOTA. Surrogate Based Optimisation is a useful and powerful way of reducing time and costs in hull shape optimisation. It allows the employ of exploratory data analysis techniques and machine learning methods to get more insight, discover hidden patterns, and detect anomalies in the design space. However, an SBO consists of several interconnected steps, and each increases complexity and uncertainty to the overall process.

The main goal is the investigation of the effects of different surrogate models on ship geometry optimisation from a resistance point of view. Also, different sampling plans, infill techniques, and optimisation algorithms are analysed and compared, as elemental steps for an SBO. This is done by the use of test functions that emulate the problem of ship optimisation. Furthermore, the intent is to try and list general guidelines to follow when building up an SBO routine with CFD simulations involved, with a focus in the marine field.

This research led to the assembly of Halton sampling sequence, Kriging meta-model, Expected Improvement function, Genetic Algorithms, and Pattern Search tools to implement a working SBO routine. This routine is used to demonstrate the success of the SBO method for a Hull Vane and an aft ship optimisations. Moreover, it is used to validate the research outcomes of this thesis and to prove that this work can be safely used for every-day commercial work.

The two design applications provided a resistance reduction, with respect to a benchmark hull, of about 19% and 7%. Thus not only they were successful, but also they were excellent examples to show how a surrogate model and its correct visualisation can give the Naval Architect the right tools to critically analyse the results, gain more understanding on the problem, spot and correct anomalies, and provide creative solutions to a client.

Acknowledgement

I would like to thank Ido Akkerman for his guidance and support during these nine months of the thesis. Thank you for always being open to discussion and confront. And thank you for helping me to become a critical thinker and a better researcher. Moreover, I would like to express my gratitude to Jerry Westerweel and Robert Hekkenberg, who accepted to be part of the thesis exam committee.

A special thank goes to Vincent, who was always patient, helpful, and available to read my work. I really enjoyed all the time spent in analysing results, issues together and in thinking creatively of how to solve them. Also, I wish to thank Niels and Perry, for allowing me to work for Van Oossanen Naval Architects. Thank you for all the resources I had available, for the possibility to do research on an exciting topic, and for the friendly and safe environment provided, even with a pandemic ongoing. Finally, a kind thanks goes to everyone in the company who greeted me with a "goedemorgen" and a "tot morgen" every day.

Most importantly, a huge thank you is for who never left my side all these years. For my parents above all, my sisters, Francesco, and my sweet grandpa, who I know would do miracles to come and celebrate with me my achievement in person. I could not have done it without you, your love, your support. Thank you for never stop believing in me and in my passion for sailing, even when you had no idea what I was studying. And thank you because you taught me that it is always worth it to dream big and to wish for more. You see, I have done it!

Contents

Figures

Tables

BACKGROUND AND CONTEXT

Introduction 1

This document is the report resulting from nine months of work at Van Oossanen Naval Architects company. It proposes a method for a Surrogate Based Optimisation (SBO) using the open-source software DAKOTA. The routine was studied to suit the requirements and the needs of Van Oossanen Naval Architects. Also, the intent is to improve the routine that the company has used in the past ten years. However, the procedure described can be useful for many different applications in different engineering fields.

The Surrogate Based Optimisation method consists of constructing a mathematical model (also known as a surrogate, response surface, meta-model, emulator) from a limited number of observations, also called points (CFD simulations, in our case). After building the surrogate, the optimisation can be performed at this level. Surrogate Based Optimisations give a solid global understanding of the problem; They allow the use of exploratory data analysis techniques and machine learning methods to get more awareness, discover hidden patterns, and detect irregularities in the design space.

1.1 Problem definition

The design and optimisation of hull shapes can be costly and challenging. The introduction of surrogate models is a useful and powerful way of reducing the costs because only a limited amount of simulations are necessary to build the approximate surrogate surface. However, the use of meta-models can also introduce errors in the solution. Moreover, several other steps and algorithms are added to the optimisation process, increasing both complexity and uncertainty. To understand which steps are involved and how they are connected, see Figure 2.1. This procedure has no certainty of convergence, and the result may be, in fact, not reducing the overall hull resistance.

Therefore, the main goal of this thesis is to investigate the effects of different meta-models on ship optimisation. Secondly, it will be necessary to select a suitable optimisation algorithm and a design space exploration plan (also called a sampling plan) to complete the SBO. This second task is critical because all steps, as we can fee again from Figure 2.1, are deeply interconnected, and misbehaviour of one influences the performance of the others. The intention is to keep uncertainties as low as possible to enhance the robustness, stability, and accuracy of the procedure.

Then, it will be necessary to test the software tools and the algorithms selected by implement a working routine. The starting point is a single initial hull geometry and the definition of parameters that

will vary the shape with specific boundaries as input. Afterwards, the routine will allow the user to generate an optimal design with minimal resistance. Finally, the intent is to improve and update Van Oossanen previous routine. In particular, the new one developed in the context of this thesis must be easier to use, cheaper to compute, and more stable.

1.2 Resources

In order to achieve the objectives of this thesis, resources from Van Oossanen N.A. as well as other third-party software tools will be used. In particular, the resistance values of the points that construct the surrogate model surface are generated via Computational Fluid Dynamics (CFD) simulations. The codes currently in use in the company were already validated and verified, and they are used every day for commercial projects. Because of this and since the main scope of this thesis is not to focus on the simulations but rather on the analysis of their results, we will consider no other options. Thus, for this thesis, the following tools are used:

Rhinoceros - Python. Rhinoceros is a commercial 3D computer graphics and computer-aided design (CAD) application software. In order to automatically generate a geometry it is coupled with the programming language Python;

Cadfix is a software solution for CAD model translation, repair, healing, defeaturing and simplification for fixing small gaps or errors in the geometry. It is used to export a Parasolid file readable by Hexpress;

Hexpress is a parallel unstructured hex dominant meshing tool for complex and unclean geometries;

FINE/Marine is a CFD software for naval architects and marine engineers;

Van Oossanen N.A. computer cluster: for the simulation execution.

Several options were taken into account to find the best toolkit that will allow both the test of different meta-models and their performance, as well as the construction of a full working routine.

Many software products are available that can perform a Surrogate Based Optimisation. Table 1.1 lists some of them and their capabilities. The most recent and updated analysis has been carried out by Viana et al. (2014) [96] , and the software products are presented in alphabetical order. It is not a complete list of all available software products. For example, Fine Design is not listed here but was also taken into account. Furthermore, it does not contain the complete list of capabilities of each tool, but rather a brief introduction to software available.

[96]: Viana *et al.* (2014), 'Metamodeling in Multidisciplinary Design Optimization: How Far Have We Really Come?'

Among them, **DAKOTA** was selected. This choice is because it is free; also the meta-model capabilities are higher than most of the other tools, as shown in the table, and because the company had a strong preference for it. The wide e variety of algorithms and options could

Table 1.1: Commercial software meta-modelling and optimisation capabilities. Analysis performed by Viana et al. (2014)[96]

Software product	Meta-model capabilities	Optimisation capabilities
BOSS/Quattro (LMS International) https://www.nafems.org/join/directory/vendors/siemens-boss-quattro-lms/	Least-squares regression for polynomials and posynomials, radial-basis functions, neural networks, kriging	Gradient-based optimisation, surrogate-based optimisation genetic algorithm, multi-objective optimisation, probabilistic optimisation
DAKOTA (Sandia National laboratories) https://dakota.sandia.gov/	Taylor series approximation, least-squares regression for polynomials, moving least squares, neural networks, kriging, radial-basis functions, multipoint approximations, multi-fidelity modelling, multivariate adaptive regression splines	Large variety of methods, including surrogate-based optimisation, gradient-based optimisation, evolutionary optimisation, multi-objective, probabilistic optimisation
Hyperstudy (Altair Engineering) https://www.altair.com/	Least-squares regression for polynomials, moving Least-squares method for polynomials, kriging, radial-basic functions	Surrogate-based optimisation, gradient-based optimisation, genetic algorithm, probabilistic optimisation
iSight (Dassault Systemes, formerly Engineous Software) http://www.engineous.com	Least-squares regression for polynomials, Taylor series approximation, radial-basis functions, neural networks, kriging, variable-complexity modelling	Surrogate-based optimisation, gradient-based optimisation, genetic algorithm, simulated annealing, probabilistic optimisation, multi-objective optimisation
LS-OPT (Livermore Software Technology Corporation) http://www.lstc.com/products/ls-opt	Least-squares regression for polynomials	Surrogate-based optimisation, gradient-based optimisation, probabilistic optimisation, multi-objective optimisation
modeFRONTIER (Esteco) http://www.esteco.it	Least-squares regression for polynomials, K-nearest interpolation, kriging, Bayesian regression, neural networks	Surrogate-based optimisation, gradient-based optimisation, genetic algorithm, simulated annealing, particle swarm optimisation, evolution strategies, probabilistic optimisation, multi-objective optimisation
OPTIMUS (Noesis Solutions) http://www.noesissolutions.com	Least-squares regression for polynomials, radial-basis functions, kriging, user-defined models, AIC methodology to find model terms	Surrogate-based optimisation, gradient-based optimisation, differential evolution, self-adaptive evolution, simulated annealing, probabilistic optimisation, multi-objective optimisation, user-defined optimiser
VisualDOC (Vanderplaats Research and Development, Inc.) http://www.vrand.com	Least-squares regression for polynomials	Surrogate-based optimisation, gradient-based optimisation, genetic algorithm, particle swarm optimisation, probabilistic optimisation, multi-objective optimisation

be overwhelming and result in less user-friendliness. However, on the other hand, DAKOTA also guarantees certain advantages:

- First of all, it is open source. Also, it is quite popular; this means one can rely on community discussion forums for support;
- Then, it can be easily coupled with other software or packages, and it is fully scriptable. Figure 1.1 represents the interface between Dakota and other simulation codes;
- Also, it has restarting capabilities and abilities to capture simulation failures;
- Finally, it can run parallel computations, also from desktop to clusters.

Design and Analysis toolKit for Optimisation and Terascale Applications (DAKOTA) is a C++ based toolkit that provides an extensible platform for performing several tasks. The ruling classes available are optimisation, uncertainty quantification, parameter study, design of experiments, surrogate modelling, and sensitivity analysis on supercomputers. For more details see the Developers [2], the Reference [3], the Theory [4], and the User [5] manuals.

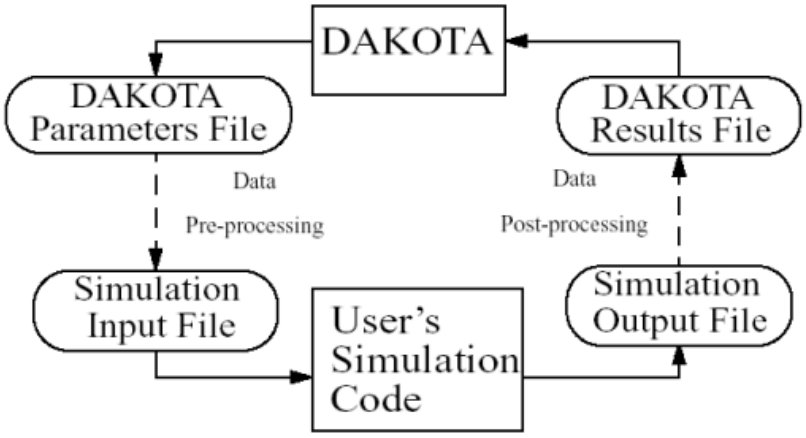


Figure 1.1: The "black box" interface between Dakota and a user-supplied simulation code [5]

DAKOTA also includes several related advanced capabilities such as nested models (layering one method over another) and parallel computing. It uses as input only one file that contains all the problem information, grouped in six blocks: environment, method, model, variables, interface, responses. There is a relationship that ties the boxes together, which we can see from Figure 1.2 for a simple study: during each iteration, a method block requests a variable-to-response mapping, which the model fulfils through an interface. More advanced cases are also possible.

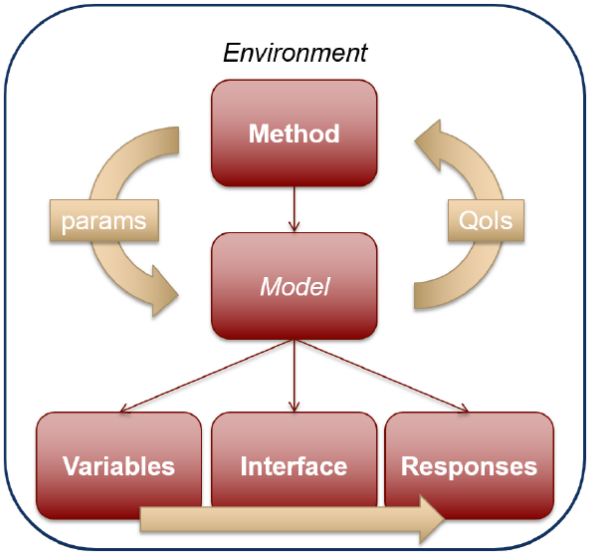


Figure 1.2: Relationship between the six DAKOTA input blocks, for a simple study [5]

1.3 Method

First, an extensive literature review was carried out on the topic and resulted in Chapter 3. It covers the Surrogate Based Optimisation topic in general and in naval architecture specifically. A historical overview was performed to demonstrate the relevance and uniqueness of this work. Also, since this thesis is completed at Van Oossanen N.A., a

careful study of the company knowledge and project development level on the topic of SBO was produced.

Second, the central research part can start. Before that, some time is needed to get to know all the tools and software, in particular, DAKOTA. Then, in order to investigate the effects of different surrogate models on ship optimisation, few carefully selected meta-models will be studied and compared. The comparison will be performed both from a literature point of view, but also with the use of test functions that will emulate the problem of ship optimisation. These analytical functions have the advantage that can give a fictional resistance value in a few seconds and not in several hours, allowing multiple tests. Another advantage is that the exact values of the function at every point are known. The functions will be chosen after careful study of typical functions used for optimisations tests in literature and after an analysis of typical surrogate responses previously generated by Van Oossanen N.A.. Afterwards, since a Surrogate Based Optimisation is composed of several connected steps (see Figure 2.1), other tests will have to be performed on sampling plans, optimisation algorithms, and refinement techniques. The tests will be carried out with the use of different DAKOTA toolkit codes specially written for each application.

Third, some time will be spent in writing the DAKOTA - Python code that will put all the pieces together to generate a working routine. Python will be explicitly used to couple DAKOTA toolkit with all the external software necessary for CFD simulations (User's Simulation Code in Figure 1.1). DAKOTA will take care of the selection of points to test, of generating the surrogate surface and of finding the best geometry. In the end, this routine will have to be tested with CFD simulations representing a hull optimisation. The number of tests is fixed to two because of time limits.

1.4 Thesis structure

This thesis is structured in various parts, each dealing with an aspect of the work conducted during the nine months of internship at Van Oossanen Naval Architects.

Part I is an introduction to the thesis. First, Surrogate Based Optimisation is presented, with its workflow, its sub-topics, and its issues (Section 2.1). Then, a historical outline of SBO and its background is given (Section 2.2). It follows an overview of Van Oossanen N.A., with emphasis on the work they carried out in the field of SBO in the past decade (Section 2.3). Finally, Chapter 3 is dedicated to showing all the literature review conducted for this thesis. It comprises four parts: the first deals with the topic of Surrogate Based Optimisation in Naval Architecture (Section 3.1) and the following three deal each with one of the SBO sub-topics. They are optimisation (Section 3.2), design space exploration (Section 3.3), and surrogate models (Section 3.4).

Part II describes the main research part of this thesis. Chapters Chapter 4 for optimisation, Chapter 5 for design space exploration

(both sampling and infill sampling plans), and Chapter 6 for surrogate models include mathematical descriptions of algorithms and methods used for the thesis. For every technique advantages, drawbacks, limitations, frameworks, and compatibility with the optimisation of hull shapes typically conducted at Van Oossanen N.A. are described. The end of this part shows the analysis and comparison of the methods performed with DAKOTA toolkit on several test functions. This Chapter 7 ends with an explanation of which settings were chosen to conduct a successful SBO.

Part III is the description of the two design applications implemented to validate the routine and demonstrate its improvements to the previous routine the company was using. The first one is a case of Hull Vane® optimisation, while the second is a case of resistance reduction obtained by modifying the aft ship geometry.

Part IV finally gives some conclusions and final considerations, as well as possible future improvements and some critical comments.

SBO: what it is and why use it 2

First, in Section 2.1, the process of Surrogate Based Optimisation is presented with its workflow and its main issues. To understand what it is and how it works, we have to give an introduction of every major step involved: sampling criteria for consequent CFD or experimental tests, surrogate models, and optimisation tools. In this section, we are giving only an overview, for more details see subsequent parts and chapters.

Second, in Section 2.2, a general and historical overview of ship design optimisation is given. We will see how in the past years, from a standard design spiral process, Naval Architects have shifted to and optimised design process where resistance is calculated through CFD tools.

Finally, in Section 2.3, the last section is dedicated to showing how Van Oossanen N.A. previously dealt with the topic of SBO. We will see the steps done in time to improve the optimisation routine. Furthermore, we want to demonstrate that this work is unique and an improvement compared to what the company did before.

2.1 SBO Workflow overview

The necessary steps of the optimisation process for hull shapes are illustrated in the flowchart in Figure 2.1. Each stage is also related to the chapter that describes it.

Firstly, the Naval Architect must choose one input geometry to analyse and optimise, also defining how many and which parameters vary in it. Furthermore, boundaries on the parameters (such as maximum bulb length) and constraints for the geometry (like fixed displacement in the resulting altered geometry) must be formulated, in order to avoid non-feasible results.

Secondly, by sampling the design space, several possible candidates are generated. The design space is the domain representing the multidimensional combination and interaction of feasible input parameters. Sampling means selecting several sets of parameter combinations within the boundaries of the domain and evaluate them with an exact model (CFD). Each sample point, which represents a new geometry that is altered to fit the new parameters, is meshed and solved. A true resistance value for each point is calculated.

Following this, a suitable surrogate model is fitted to the available exact resistance data resulting from the numerical simulations. The fit represents an approximated resistance function that varies for each selected variable. The model must be estimated through parameter

tuning and validated by checking its accuracy. If it is not enough, one way to improve it is to add infill points to the process and update the surrogate model. It means solving resistance for new hull geometries. This step is essential is because the initial design selection will almost inevitably miss certain features of the surrogate surface to analyse and this way areas of interest (near minimum points, for example) can be better examined. This loop is carried out many times until the surrogate fits its purpose or perhaps the available budget of computing effort has been exhausted.

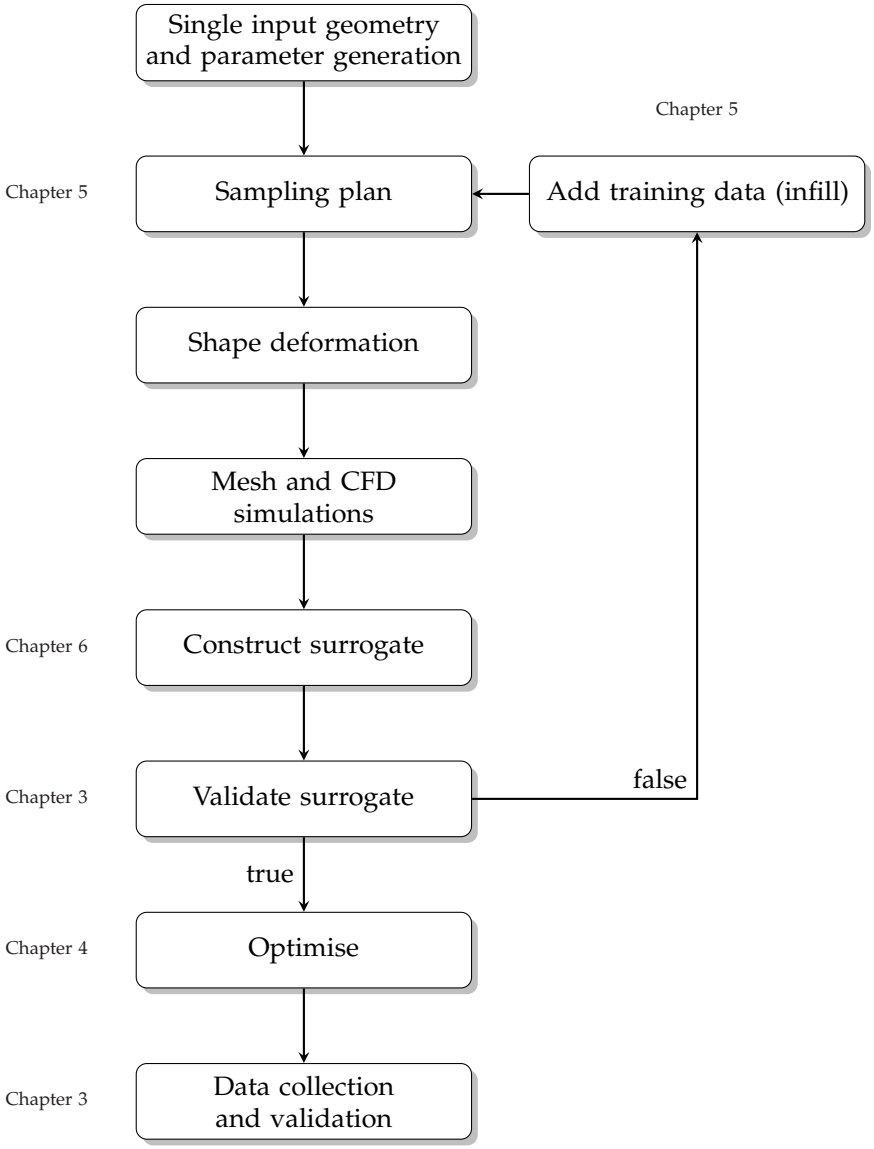


Figure 2.1: Surrogate based optimisation workflow

Then, the constructed meta-model fit can be used for optimisation purposes. A significant advantage of using a surrogate model is that the search for the point of lowest resistance can be done on the approximate continuous surface newly generated. This way, a high number of points can be tested by optimisation algorithms.

Finally, an essential part of the process is the possibility for the Naval Architect to gain insight into the problem. A correct, exhaustive, and precise data visualisation will give the user the right tools to judge

the optimisation routine accuracy and uncertainty. Also, this is the phase where the initial design is compared to the optimised one.

2.1.1 SBO workflow issues

As already established, the Surrogate Based Optimisation has proven to be quite successful; however, each application is different, and some issues arise when it comes to tuning the routine for the purpose required in this specific thesis. Here is a list of questions and issues that are important to pose when dealing with an SBO routine. The order in which each topic is presented is the same as in figure 2.1. The solutions proposed for these questions in this thesis are given in details in the following chapters.

Sampling plan:

- How many points are sufficient and necessary to represent the domain per variable fully? Furthermore, what is the relation between this number and the number of variables?
- Which points to chose, i.e. spread how in the domain? What does the decision depend on?
- Noise is inevitable in CFD results and thus in the sample points. How does it affect the meta-model behaviour?

Meta-model:

- Which factors influence the decision to use a surrogate model over others? Moreover, what are the factors that characterise a common case that has to be solved in Van Oossanen N.A.?
- How to assess model accuracy? Furthermore, which factors play a role in the model validation?
- What is the best way to visualise the meta-model surface, in such a way that the user can gain insight into the problem and make informed decisions?

Infill:

- How many points should be used in the refinement phase? Moreover, what is the ratio between this number and the number of initial sampling points?
- Which points to chose in the domain? How to balance a global and a local search? When to stop the search?

Optimisation:

- What method is best for the kind of surfaces that are most commonly faced? Furthermore, what influences that choice?

2.2 Historical overview

The design of ships involves a multitude of factors, including size, cost, speed, stability, seaworthiness, comfort, manoeuvrability, accommodation space, aesthetics, and each of them is traded off against each other. Traditionally, to satisfy multiple and conflicting objectives, naval

architects have proceeded through a series of iterations. This process is referred to as a design spiral and it is shown in Figure 2.2 (Evans in 1959 [20] and its improvements by Buxton in 1976 [10] and Andrews in 1981 [6]).

[20]: Evans (1959), 'Basic Design Concepts'

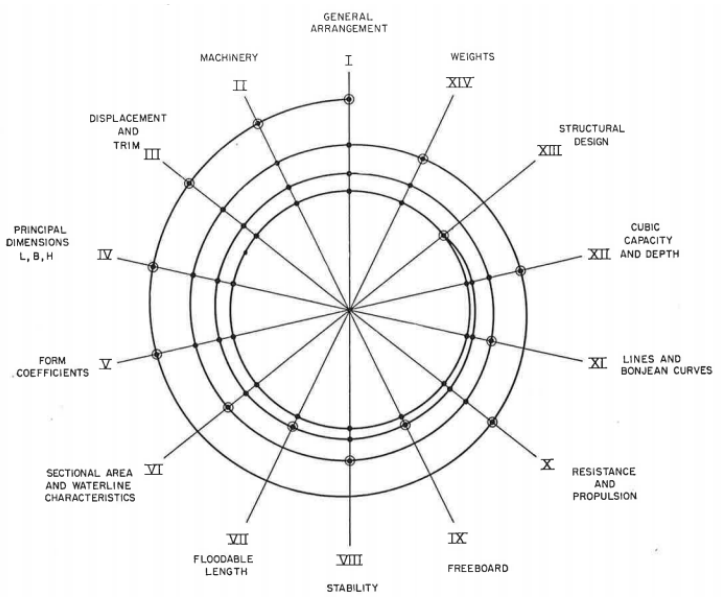


Figure 2.2: Design spiral (Evans, 1959 [20])

Designers used this process because among its advantages, we count its simplicity and computationally efficiency, its flexibility, and clarity in showing the iterative nature of the design. However, it is time-consuming, and it does not guarantee to result in an optimal design. Moreover, the possibility to be re-used for different designs is limited. Therefore, the design process has profoundly changed over the last decades, and the attention in recent years has shifted from using a converged design process to an optimised one.

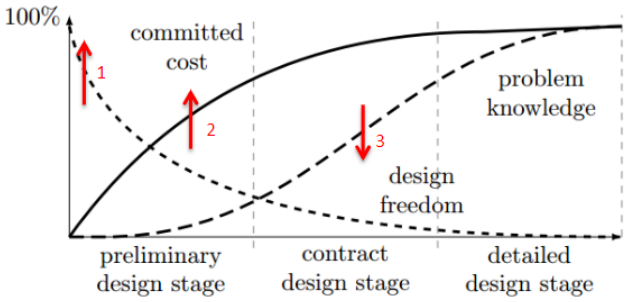


Figure 2.3: Relationship of design freedom, knowledge, and cost committed. Analysis performed by Mavris and DeLaurentis (2000) [59]

This new process helps in facing the three main challenges a naval architect has to deal with during the design phases, as we can see from Figure 2.3:

1. Design freedom is (perhaps too) high upfront in the preliminary design stage;
2. Many costs get locked-in quite early;
3. The important decisions have to be taken when the understanding of the design problem is still minimal.

Even though this new approach improves the work of naval architects, it also gives rise to new issues. In particular, it needs new rules to

deal with conflicting objectives and define priorities when designing a complex item such as a boat. For simplicity, and because it is one of the biggest parts in the design of a vessel, here we focus our attention solely on the estimate of resistance and propulsion.

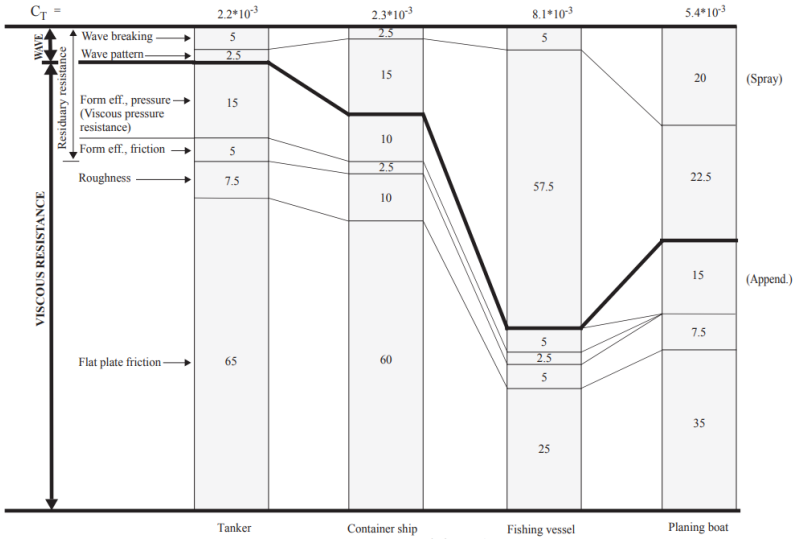


Figure 2.4: Resistance components for four vessels (%). Analysis performed by Larsson and Raven (2010) [53]

Total resistance is the sum of many different components, with the wave and viscous resistance being the main ones. The former occurs because the hull generates waves, transferring energy away, while the latter is due to frictional forces between hull and water. We can see the components of resistance for four different kinds of vessels in Figure 2.4. As the total resistance estimate largely depends on the shape of the hull and its wetted surface, it also directly affects or is affected by most of the other voices in the design spiral in Figure 2.2, namely Principal Dimensions, Form Coefficients, Sectional Area and Waterline Characteristics, Stability, Weights. Again, for the sake of simplicity, in this thesis, we will deal with the optimisation of the hull shape only taking into account the cost and time aspects of the design phase (but not of the production). Also, some preliminary work on the hull is considered to have been done, in particular dimensions and weights. Stability and sea-keeping analysis are not directly treated, even though they could be added to the routine in the future.

To get an estimated value of the total resistance, traditionally towing tank experiments (Figure 2.5) are used. The advantage is that it is easy to test multiple velocities for one hull geometry. However, the number of prototypes and conditions to test in the tank are limited. Also, this method requires an in-depth knowledge of the problem to choose which configurations to analyse. Moreover, it is subjected to random errors; thus, it is not a deterministic process. Nowadays, the high costs of physical experiments and field tests can be reduced by the use of simulation software able to model complex problems.

Computational Fluid Dynamics (CFD) is a numerical way to solve the Navier Stokes equations and deal with problems involving fluid flows, hence also able to calculate the resistance of a vessel (Figure 2.6). It was first developed in car and plane industry in the late 1960s.

Figure 2.5: Example of towing tank test

Afterwards, the marine industry also started making use of it firstly thanks to sailing competitions, namely America's Cup.

In the context of optimisation, a large number of different designs have to be tested. Therefore, CFD methods are more efficient than towing tank experiments because they do not involve the construction of models to test, thus allowing an easy test of multiple geometries. The hull is generated with the use of Computer Aided Engineering (CAE) and Computer Aided Design (CAD); thus, any modification can just be made on a computer. Different methods with different assumptions have been developed. They are here described in order of simplicity, computational cost, and accuracy:

Reynolds Averaged Navier Stokes (RANS) where the equations are averaged and closed by using a turbulence model;

Large Eddy Simulations (LES) in which the most computationally expensive smallest turbulence vortex lengths are ignored;

Direct Numerical Simulations (DNS) where Navier Stokes equations are fully numerically resolved.

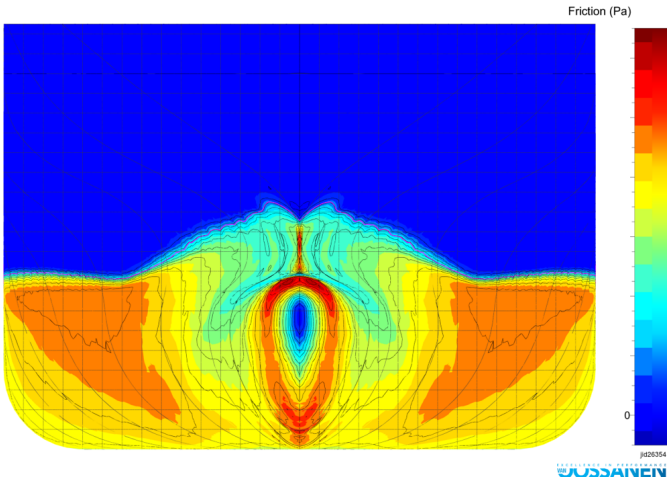


Figure 2.6: Example of CFD simulations results: friction values obtained on the front part of a hull surface. Courtesy of Van Oossanen N.A.

In particular, in the context of this thesis, an (Unsteady) Reynolds Averaged Navier Stokes (U)RANS equations code will be used. These equations need a closure model for which the two-equation $k-\omega$ SST Menter turbulence model was used. Even though it gives only a time-averaged mean value for resistance, it requires the lowest computational effort, while still giving acceptable results. Nonetheless, when it comes to search for the optimal hull shape, and thus perform a high

number of simulations for different hull shapes or Froude numbers, this method is still too costly. One simulation takes around 20 hours.[1] For a two-variable optimisation problem, assuming an average of 60 computations is needed for the optimisation, the total computational time would be around 50 days, which is not acceptable in practice for a client.

1: Data from Van Oossanen Naval Architects regarding the simulation of one hull.

Consequently, the idea is to compute only a limited and well-selected number of CFD simulations and use the results to construct a meta-model that approximates the infinite possibilities between the few tested. This way, time and cost are reduced; moreover, new geometry possibilities not previously considered are evaluated.

Finally, with the use of meta-models, the following step is to automate the process of finding the optimal shape. A routine can be built and will be further on called Surrogate Based Optimisation (SBO).

2.3 SBO in Van Oossanen

The topic of Surrogate Based Optimisation is not new in the company; in fact, from 2009 Van Oossanen N.A. can successfully perform SBO routines. However, the features have changed over the years, with the help of several students and interns. Table 2.1 summarises the main aspects of the work done so far, and the improvements expected to achieve during the nine months for this thesis. We can see on the left column listed the different steps of the SBO as described in Figure 2.1. On top, there are the names of who worked on different projects during the years. The table is filled with sampling methods names (LHS, Halton), meta-model names (KG and CO-KG), optimisation algorithm names (GA, BFGS, PS), and infill technique names (EI) that will be described more in detail later on.

First, Zaaijer, in his M.Sc. thesis (2009) [101] described a routine that involved only a 2D approach, mainly due to computation capabilities. Nevertheless, all the parts described in the workflow in figure 2.1 are already in place. Secondly, Delivre in his B.Sc. thesis (2014) [16] adopted a 3D approach and introduced the use of R language to deal with the routine. Moreover, he made some small changes in the choices of models and methods to use: Halton sequence replaces LHS sampling method, a variant of simple Kriging method that involves multi-fidelity data is adopted, and Broyden Fletcher Gordfarb Shanno (BFGS) local search is added to the optimisation algorithm. However, these changes were entirely based on literature review, thus missing a rigorous test phase. Thirdly, Cointe in his internship (2016) [13] mostly worked on multi-fidelity data. The conclusion was that no real advantage is achieved in having less expensive and accurate CFD results, therefore abandoning co-Kriging method. Finally, Bihan in his internship (2018) [54] added the possibility to use a two-objective optimisation. He also reintroduced the application of Expected Improvement (EI) for infill sampling.

[16]: Delivré (2014), 'Optimisation de carène par modèle de substitution'

Table 2.1: Comparison between surrogate models. Analysis performed by Q. Delivré (2014)[16].

	ZAAIJER (2009)	**DELIVRE** (2014)	**COINTE** (2016)	**LE BIHAN** (2018)	**CASELLA** (2020)
Approach	2-Dimensions	3-Dimensions		2-Objective	mid-routine checks
Main toolkit		R language			DAKOTA toolkit
Geometry	CAD Rhino - Hexpress	Rhino Python - coupling			
CFD	FINE/Marine - RANS solver		Mesh quality CFD iterations Low fidelity data		FINE/Marine and OpenFOAM
Sampling	LHS	Samping study (HALTON)		Number of sampling points	Sampling rigorous study (HALTON) Number of samples rigorous study (12d - 15d)
Surrogate	Kriging	Surrogate study (CO-KRIGING)	Kriging/ Co-Kriging comparison	Kriging back	Surrogate rigorous quality study (KG) Noise rigorous study
Optimisation	GA	GA-BFGS			Optimisation rigorous study (GA-PS)
Infill		EI	EI Deleted	EI back	Infill rigorous study (EI)

The main change carried out in this thesis concern the switch from R language to DAKOTA toolkit. This new solution improves user-friendliness. Also, the most significant part of this thesis aims to do a more detailed and rigorous study on sampling and surrogate methods, on optimisation and infill algorithms. The more recent advances in SBO techniques and the specific capabilities of DAKOTA are investigated. Furthermore, to improve the chances of success and reduce the possibility of non-convergence of the SBO, the routine will be split into a few automatic steps. After each step, visualisation tools will help the user to check the results before going to the next phase. The reason for this choice is that it is a strict requirement of the company to avoid unnecessary CFD simulations. Lastly, if time allows, it will be added the possibility to use the open-source OpenFOAM CFD software. This new feature has the aim to give Van Oossanen N.A. the possibility to lower down the expenses incurred when optimising with CFD tools that require a licence. However, in the context of this thesis, the simulations will be performed with FINE/Marine in order to be able to compare new results from old ones and validate the routine.

Literature Review 3

This chapter describes the literature review conducted for this thesis. To start, in Section 3.1, an overview of authors that worked on Surrogate Based Optimisation applied to different engineering problems is given. In particular, more attention is paid on works related to Naval Architecture, to ship resistance reduction problems, and to DAKOTA applications. Afterwards, each sub-topic is analysed more in detail.

First, Section 3.2 gives an overview of what is an optimisation from a mathematical point of view, what is found in the literature to be useful when comparing different optimisation algorithms, and also an analysis of different methods and their classification. In the end, it is specified which algorithms are most commonly used in literature for similar applications as the one here presented. Also, the reader will understand which algorithms are expected to pursue and the reasons why they look more promising than others.

Second, in Section 3.3, both the initial sampling choice as well as the infill sampling criteria (how to improve the accuracy of the surrogate adding points to the initial sampling plan) are going to be analysed. In particular, the influence of sample size and noise is going to be discussed. Also, we need to understand what are the characteristics that make a method or another better in selecting the point location. Finally, it is explained which methods are most commonly used in literature and the reasons behind the decisions made for this thesis.

Finally, in Section 3.4, a general discussion and overview of meta-models are given. Also, it is explained which models are analysed in the central research part of this thesis and why. Furthermore, a section is dedicated to explaining the most common and appealing techniques found in the literature to assess the model's accuracy. In the end, we will see which visualisation tools and methods were found to be more effective in literature.

3.1 SBO in literature

SBO has been widely investigated for many marine applications. Barthelemy and Haftka reviewed in 1993 the main approximation concepts in applications of non-linear programming to structural optimisation [8]; A. Mason et al. between 2004 and 2010 investigated the optimisation of sailing performance. The authors first focused on the applicability of Artificial Neural Networks (ANN) to a catamaran resistance reduction [15] [57], then worked on ANN meta-model to automate the optimisation of International America's Cup Class (IACC) yachts. Both resistance, with CFD, and performance, with Velocity

Prediction Program (VPP) were considered [58] [55] [56] . Also, among others, in 2013 Iuliano and Quagliarella [41], and Ulaganathan and Asproulis [92] used surrogate models in foil shape optimisation; Vesting and Bensow looked into propeller performance evaluation [94]; wave propagation was investigated by Roselli et al. [76], and sailing dagger-boards were optimised by Guerrero et al. [27].

[56]: Mason (2010), 'Stochastic Optimisation of America's Cup Class Yachts'

Examples of resistance reduction in naval architecture include a bulb shape optimisation performed by Delivre in 2014 [16], by Cointe in 2016 [13], and by Cominetti and Guerrero in 2017 [14] [28] . Also, Couser in 2004 [15] worked on reducing a catamaran resistance employing a Surrogate Based Optimisation.

[28]: Guerrero *et al.* (2018), 'Surrogate-Based Optimization Using an Open-Source Framework: The Bulbous Bow Shape Optimization Case'

Finally, several companies and institutes are currently using or studying an SBO with DAKOTA toolkit. In particular, CINECA through Spisso [89], Wolfdynamics with Guerrero [27] [25], and MARIN mainly thanks to the work of Scholcz and Raven [72] [73] [81] are developing a routine for shape hull optimisation with DAKOTA and OpenFOAM, or are providing consulting on it. Besides, among the universities that investigate SBO in marine applications, we count Universitá degli studi di Genova (UniGe), Italy, with the MSc thesis of Cominetti in 2017 [14] followed by a publication by Guerrero [28] on the same topic. Also, Chalmers University, in Sweden, published two MSc theses of Jareteg in 2014 [42] and Ohm in 2017 [65] on the matter.

[81]: Scholcz *et al.* (2019), 'Surrogate-Based Multi-Objective Optimisation for Powering and Seakeeping'

All these papers have proven that Surrogate Based Optimisation is indeed a topic of great interest, mainly when applied to Naval Architecture. All authors conclude that the reduction of time and costs are significant with respect to a traditional optimisation. Thus, it is clear that this thesis presents a solution to the hull geometry optimisation that is in line with current research. Also, the application of DAKOTA toolkit in this field was proven successful and the focus of recent studies by several institutes. However, if we would investigate more in detail this small list of examples, see Tables 3.1 and 3.2, their choice of surrogate model, their choice of algorithms and workflows, we could see that all differ. It is thus clear that no method was outstanding the others for most applications. No standard rules were ever proposed for SBO because too many variables can affect them. Also, a wide variety of applications are possible within the small field of Naval Architecture. Moreover, from the table, it would also seem that there is not one only successful way to proceed.

Therefore the gap that this thesis is trying to fill in is the investigation of SBO methods and algorithms when it comes to applying it to a specific purpose for a specific company: Hull Vane® [3] , aft ship, and bulb optimisations at Van Oossanen N.A.. Furthermore, to make this work useful for the future researcher, the intent is to try and list general guidelines to follow when building up an SBO routine with CFD simulations involved.

3: See more information on: https://www.hullvane.com/

Table 3.1: Summary of SBO references. Part I

Work by	Type of problem	Variables	Objectives	Sampling	Surrogate	Optimisation	Infill
Simpson et al. (2000)	Structural analysis of a two-member frame	3 [a]	1 (Minimum frame volume)	Hammersley, LHS, OA, random, UD (9, 16, 25, 32, 49, 64)	KG, RBF, second order response surface, MARS	-	-
Couser, Mason et al. (2004)	Catamaran resistance prediction	3 [b]	1 (Residuary resistance coefficient)	Experimental available points (68)	ANN	-	-
Mason et al. (2005)	Preliminary prediction of vessel resistance	3 [c]	1 (Residuary resistance coefficient)	Experimental available points (10)	ANN	GA	-
Mason (2010)	IACC yacht optimisation	5 [d]	2 (Resistance via CFD, performance via VPP and RMP)	Pseudo-Monte Carlo (25)	ANN	GA	7 random points and 7 UD points
Iuliano and Quagliarella (2013)	Foil shape optimisation	16	1 (lift, drag coefficients ratio)	LHS (50)	KG, polynomial regression		EI
Ulaganathan and Asproulis (2013)	Foil shape optimisation	11 [e]	1 (lift coefficient)	Hammersley (50)	KG	GA	-
Vesting and Bensow (2014)	propeller performance evaluation	6 [f]	3 (Propeller efficiency, maximum pressure pulses, maximum cavity volume)	Sobol algorithm (200)	KG, ANN	EA / local tangent search	-

[a] Frame width, height and wall thickness
[b] Slenderness, breadth - draught ratio and separation - length ratio
[c] Breadth - draught ratio, slenderness, volume
[d] Maximum beam of waterline, prismatic coefficient, midship area coefficient, non-dimensional longitudinal centre of buoyancy, flare
[e] Leading edge radius, upper crest point, lower crest point, position of upper crest, position of lower crest, upper crest curvature, lower crest curvature, trailing edge thickness, trailing edge offset, trailing edge wedge angle, trailing edge direction angle
[f] Chord, camber, pitch, rake, skew, thickness

Table 3.2: Summary of SBO references. Part II

Work by	Type of problem	Variables	Objectives	Sampling	Surrogate	Optimisation	Infill
Delivre (2014)	Bulb shape optimisation	4 [a]	1 (Total resistance)	Halton (50)	Co-KG	GA-BFGS	EI
Guerrero et al. (2015)	Sailing yacht daggerboard optimisation	12	2 (Drag and vertical force)	GA	*not specified	GA	GA
Guerrero et al. (2018)	Bulbous bow shape optimisation	2 [b]	1 (Total hull resistance)	Full-factorial (25)	KG	GA	Manually
Raven and Scholcz (2017)	Bulb shape optimisation	5	2 (Wave resistance at two different speeds)	LHS (127)	KG	GA	SBG
Raven and Scholcz (2019)	Ship stern shape optimisation with DAKOTA	3 [c]	1 (Hull Resistance)	LHS (150 low fidelity and 32 high fidelity)	Co-KG	GA	-
Ohm (2017)	Turbine performance optimisation with DAKOTA	5	2 (Efficiency and pressure drop)	LHS (51)	KG	EGO	EI (61)
Roselli et al. (2018)	Wave propagation	5	2 (Space and time wave characteristics)	Sobol Algorithm (320)	KG	GA	-

[a] Bulb length, height, angle, width
[b] Protusion and immersion of bulb
[c] Two stern buttock shape modifications and deadrise at the stern

3.2 Optimisation

Optimisation is defined as an act, process, or methodology of making something (such a design, system, or decision) as fully perfect, functional, or effective as possible [4] . An optimisation problem is mathematically stated as:

4: Available at: Merriam-Webster.com Dictionary, https://www.merriam-webster.com/dictionary/optimization.

$$minimise \quad f(\boldsymbol{x}) \tag{3.1}$$

$$with \quad \boldsymbol{x} = (x_1, x_2, \cdots, x_n)^T \tag{3.2}$$

$$subject\ to \quad h_i(\boldsymbol{x}) = 0, \quad \text{for } i = 1, \cdots, m \tag{3.3}$$

$$g_j(\boldsymbol{x}) \geq 0, \quad \text{for } j = 1, \cdots, p \tag{3.4}$$

$$x_{k,\,\min} \leq x_k \leq x_{k,\,\max}, \quad \text{for } k = 1, \cdots, n \tag{3.5}$$

having the following components:

- Equation 3.1 is the **objective function**, the criteria by which a solution is assessed. For example, total hull resistance;
- Equation 3.2 represents the **design variables** which uniquely describe the optimisation problem. For example, the geometry parameters used to vary the hull geometry such as the Hull Vane chord;
- Equations 3.3 (equality), 3.4 (inequality), and 3.5 (bound) are the **constraints** that define the boundaries of the feasible regions of the design space. For example, the hull displacement has to be fixed, or the Hull Vane chord cannot be too small or it is not possible to build it.

In the case of this thesis, the function f to minimise is too expensive to compute. It can be called a "Black-Box" function, as no assumption of differentiability, convexity or smoothness on the output can be made [7]. Thus an approximate meta-model surface $\widehat{f}$ is used. $\widehat{f}$ emulates the behaviour of hull resistance when varying different design variables representing a feature of the geometry such as height, width, or position of an element. It has the great advantages of being cheap to compute and continuous; however, it contains uncertainties. In fact it can be that $min(f) \neq min(\widehat{f})$. Also, at Van Oossanen N.A., simple bound constraints such as $0 \leq x_n \leq 1$ with n the number of variables are used to define the available domain.

More in general, it is possible that there is more than one objective function to optimise. It is the case when several criteria $\boldsymbol{f}(\boldsymbol{x}) = [f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \cdots, f_p(\boldsymbol{x})]^T$ characterise the performance of a design and this is called Multi-Objective Optimisation (MOO), or Multi-Disciplinary Optimisation (MDO) if objectives are of different fields. Examples in Naval Architecture field include the minimisation of hull resistance for two different Froude numbers (i.e. two different boat

[28]: Guerrero *et al.* (2018), 'Surrogate-Based Optimization Using an Open-Source Framework: The Bulbous Bow Shape Optimization Case'

speeds) [28] or the optimisation of both resistance and performance [57] [69]. It is acknowledged that a multi-objective optimisation can be of great benefit when dealing with a multitude of factors involved in the design of a vessel (see Figure 2.2). However, for the sake of simplicity, during this thesis, we are going to focus on a single-objective optimisation, namely resistance at one speed. Nevertheless, the routine will be built with the possibility of adding objectives when required.

3.2.1 Comparing methods

[7]: Audet *et al.* (2017), 'Derivative-Free and Blackbox Optimization'

In order to decide which method works best for our application, comparisons will make use of artificially generated test sets, in the form of analytical functions. Audet and Hare (2017) [7] say that when studying optimisation algorithms, the main goal is to compare the efficiency and the quality of the solution. Efficiency refers to the computational effort required to obtain a solution. Quality of the solution refers to the precision of the algorithm's final output. We will mainly focus on quality because, for our application, the CPU time required to perform an optimisation is in the range of minutes. Thus, time is negligible compared to the CFD computation one, in the range of hours. The closeness of the solution assesses the quality of optimisation found the algorithm with the true one if known.

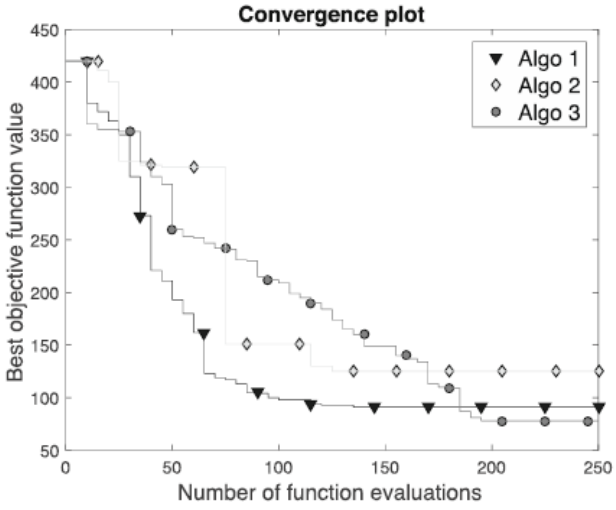


Figure 3.1: Example of a convergence plot for a given test problem and three different optimisation algorithms. From Audet and Hare (2017) [7]

Furthermore, it is crucial to monitor convergence. In optimisation, convergence is usually used to indicate that the algorithm reaches the final solution. It is necessary to note that a convergent optimisation process does not necessarily mean the global minimum (or minima) has been found [64]. A way to visualise it is with the use of convergence and trajectory plots. Figures 3.1 and 3.2 are example from [7]. In both of them, at this stage, it is not important to distinguish the difference between each Algo(rithm). The first figure visualises the performance of different optimisation methods by plotting the best possible objective function value found against the number of function evaluations used. The second, restricted to functions of two variables, is created by plotting the contour plot of the objective function (when such information is available) and then plotting the paths that connect

the points generated by each iteration of each algorithm when applied to the objective function.

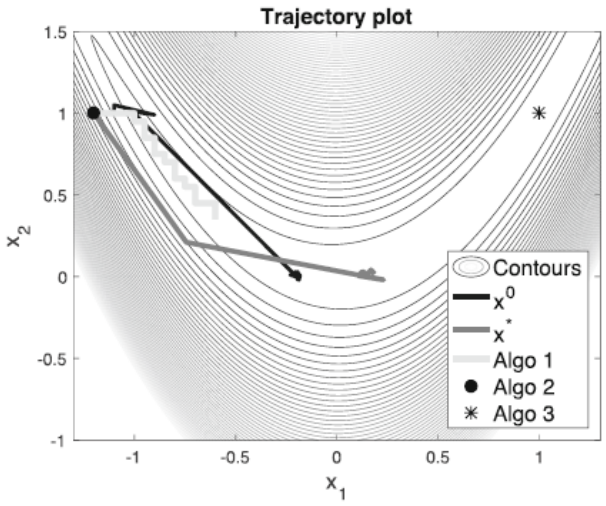


Figure 3.2: Example of a trajectory plot for a given test problem and three different optimisation algorithms. From Audet and Hare (2017) [7] [a]

[a] Attention: in the legend x^0 is the filled point, x^* is the asterisk, Algo 1,2, and 3 are the three thicker lines representing the three algorithm trajectories

Finally, another essential feature to control when testing different optimisation algorithms is the termination criteria. It must be set correctly unless the optimisation will: (i) fail to converge to a stationary solution (too loose criteria) or (ii) result in useless evaluations, thereby extra optimisation time (too tight criteria) [64]. The most common options used count: maximum number of iteration, maximum time, acceptable objective function, or objective function convergence.

3.2.2 Optimisation algorithms

The main issue about optimisation algorithms recognised for this thesis is the selection of which one is best for our application. Mainly because a high number of possibilities are available in the literature and also supported by DAKOTA toolkit; furthermore, an algorithm that is best on one problem may not be best on a different problem. The algorithm chosen for an optimisation task will largely depend on the type of the problem, the nature of an algorithm, the desired quality of solutions, the available computing resource, time limit, availability of the algorithm implementation, and the expertise of the decision-makers [100].

Optimisation algorithms can be classified in many ways, depending on the characteristics we are trying to compare [50] . In particular:

[50]: Koziel *et al.* (2013), *Surrogate-Based Modeling and Optimization: Applications in Engineering*

Gradient-based and gradient-free Methods such as Genetic Algorithm (GA) [24] are derivative-free because they only use the values of the objective. These methods show slow convergence rates for finding an optimum, and they usually require from thousand to tens-of-thousands of function evaluations. However, they can be more robust [5]. On the other hand, methods such as Quasi-Newton [9] (that uses only first-order derivatives) and Newton (that use also second-order derivatives), are usually highly efficient in term of computational time. Nevertheless, they entirely rely on gradient accuracy. In this sense, analytical gradients are

ideal, but often unavailable as in our case because they are a product of the surrogate model. Thus, with a non-smooth, discontinuous, or multi-modal function behaviour, inaccurate gradients will lead to bad search directions [5]. In conclusion, if gradient information is available, reliable, and obtainable at a reasonable cost, then gradient-based methods should be used [7].

Trajectory or population based A trajectory-based algorithm typically uses a single agent or solution point which will trace out a path as the iterations and optimisation process continue. However, population-based algorithms such as Particle Swan Optimisation (PSO) use multiple agents which will interact and trace out-multiple paths [48].

Deterministic or stochastic If an algorithm works in a mechanically deterministic manner without any random nature, it is called deterministic. For such an algorithm, it will reach the same final solution if we start with the same initial point. An example is the Nelder-Mead downhill simplex method [63]. On the other hand, if there is some randomness in the algorithm, the algorithm will usually reach a different point every time we run the algorithm, even though we start with the same initial point. Genetic algorithms are a good example of this category.

Local and global Local search algorithms typically converge towards a local optimum, not necessarily (often not) the global optimum, and such algorithms are often deterministic and cannot escape local optima [7]. Hooke-Jeeves pattern search algorithm [38] is amongst the best known. On the other hand, we always try to find the global optimum for a given problem. If this global is robust (i.e. the optimised solution is a design that works well for certain conditions and also when the conditions fluctuate slightly [98]), it is often the best. Global Algorithms, such as Genetic Algorithms, are also preferable when the objective function is multi-modal, meaning they have more than one global minimum.

[98]: Whitfield *et al.* (1998), 'A Robust Design Methodology Suitable for Application to One-off Products'

In the case of SBO applied to Naval Architecture, most authors focused on the application of Genetic Algorithms (GA) and its variants. Examples count Guerrero et al. work in 2018 [28]. However, they clearly state that when optimisation is done on the surrogate level, any optimisation method can be used. Also researchers at MARIN [72] [81] used Genetic Algorithm (MOGA). However, they mainly focus on Multi-Objective Optimisation. Another example is Mason et al. (2005) [57] that used GA for a catamaran resistance reduction. More interestingly, authors like Vesting and Bensow (2014) [94] used a combination of two methods: Genetic Algorithm was applied to detect the region of global optima which was then used as the starting point for a local tangent search algorithm [35], to find the local optimum. At Van Oossanen N.A., also, a combination of a global gradient-free population-based method (GA) and a local gradient-based trajectory-based method (BFGS) is used.

[94]: Vesting *et al.* (2014), 'On surrogate methods in propeller optimisation'

To conclude, for this thesis, the feeling is that a combination of a local and a global algorithm is the best choice. The reason is that it

Method Classification	Desired Problem Characteristics	Applicable Methods
Gradient-Based Local	smooth; continuous variables; no constraints	optpp_cg, rol
	smooth; continuous variables; bound constraints	dot_bfgs, dot_frcg conmin_frcg, rol
	smooth; continuous variables; bound constraints, linear and nonlinear constraints	npsol_sqp, nlpql_sqp, dot_mmfd, dot_slp, dot_sqp, conmin_mfd, optpp_newton, optpp_q_newton, optpp_fd_newton, rol weighted sums (multiobjective), pareto_set strategy (multiobjective)
Gradient-Based Global	smooth; continuous variables; bound constraints, linear and nonlinear constraints	hybrid_strategy, multi_start strategy
Derivative-Free Local	nonsmooth; continuous variables; bound constraints	optpp_pds
	nonsmooth; continuous variables; bound constraints, nonlinear constraints	coliny_cobyla, coliny_pattern_search, coliny_solis_wets,
	nonsmooth; continuous variables; bound constraints, linear and nonlinear constraints	asynch_pattern_search, surrogate_based_local
	nonsmooth; continuous variables; discrete variables; bound constraints, nonlinear constraints	mesh_adaptive_search
Derivative-Free Global	nonsmooth; continuous variables; bound constraints	ncsu_direct
	nonsmooth; continuous variables; bound constraints, nonlinear constraints	coliny_direct, efficient_global
	nonsmooth; continuous variables; bound constraints, linear and nonlinear constraints	surrogate_based_global
	nonsmooth; continuous variables, discrete variables; bound constraints, nonlinear constraints	coliny_ea
	nonsmooth; continuous variables, discrete variables; bound constraints, linear and nonlinear constraints	soga, moga (multiobjective)

Table 3.3: Guidelines for optimisation method selection. From DAKOTA User's Manual [5]

is not possible to know in advance if the objective function is multimodal or not, or if it is linear and smooth or not. This way, a higher chance of capturing several possible optimum points is foreseen. In particular, for the global method, it looks like from this literature review that a Genetic Algorithm may be the best fit for this thesis. It is also implemented in DAKOTA toolkit. However, this technique requires some tuning that will need to be performed. As for the local algorithm, DAKOTA has a various number of possible choices. They are visible in Table 3.3. The two methods that look more promising, and therefore will be analysed more in detail, are the gradient-based `optpp_q_newton`, a version of the Quasi-Newton BFGS algorithm, and the derivative-free `coliny_pattern_search`.

3.3 Design space exploration

When a cheap-to-evaluate surrogate model $\widehat{f}$ has to be constructed to emulate the expensive response of some black box continuous function $f(\boldsymbol{x})$, an important part is the definition of a d-vector of design variables $\boldsymbol{x} \in D \subset \mathbb{R}^d$. In what follows we shall refer to D as the de-

sign space or design domain. The discrete observations or samples $\{\boldsymbol{x}^{(i)} \rightarrow y^{(i)} = f(\boldsymbol{x}^{(i)}) \mid i = 1, \cdots, n\}$ are defined as part of a sampling plan $\boldsymbol{X} = \{\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \cdots, \boldsymbol{x}^{(n)}\}$ [23]. Different sampling plans may result in the same surrogate model, but with different accuracy. Thus the goal is to allocate points in the design space that minimise the influence of errors on the response functions f, while allowing the designers to build meta-models more efficiently [102] and make a cheap but reliable performance prediction for any $\boldsymbol{x} \in D$.

3.3.1 Sample size

[86]: Simpson *et al.* (2001), 'Metamodels for Computer-based Engineering Design: Survey and recommendations'

Per other authors like Simpson et al. (2001) [86] , we recognised that the primary concern in the design of an experiment is its size. Intuitively, the higher the number of sampling points, the higher the quality of surrogate. In particular, the appropriate sample size depends on three main factors:

- The nature of the experiment required to obtain the points part of the sampling plan and, more specifically, the cost necessary to make each estimation. As an example, CFD simulations are costly;
- The complexity and (non-)linearity of the function to be approximated. It has been observed by Wang et al. that, after reaching a certain sample size, increasing the number of sample points does not contribute much to the approximation accuracy [97]. This characteristic is especially valid for low order functions;
- The number of design variables in the modelling problem. In particular, Forrester et al. analysis pointed out that if a certain level of prediction accurateness is achieved by selecting n points for a one-variable space, in theory, to achieve the same sample density in a d-dimensional space, n^d observations are required [23]. However, this exponential behaviour can become very expensive with just a small increase of design variables. Thus it is not feasible in the case of this thesis where containing simulation time is paramount.

The two crucial conclusions are that, firstly, a trade-off is necessary between the desire to have an accurate meta-model with many sampling points and the desire to lower down the hours necessary to obtain those same points. The second conclusion we can draw is that the number of design variables has a massive impact on the number of experiments required. It is, therefore, imperative that they are minimised [23]. Also, because we foresee that for this thesis it will not be possible to achieve n^d observations, it will become essential to develop proper visualisation tools that allow the user to grasp the trend of the objective function (resistance) over the various design variables.

As the problem of defining the sample size depends on so many variables, no author in literature has been able to identify a fixed set of rules. Nonetheless, we can cite here, as an example, the work of Simpson et al. (2001) [86]. The authors tested a structural problem with three design variables and concluded that the accuracy of several meta-models highly increased from 9 to 25 sample points and kept (slightly)

increasing up to 64. Also, other authors like Jones et al. (1998) [47] found out that about 10d points build with a space-filling method are needed. However, this conclusion was solely based on their experience, and they also explained that they found it convenient sometimes to deviate slightly from this rule. Within Van Oossanen N.A., Le Bihan [54] worked on the definition of the sample size. His recommendation is to work with approximately 25 points for three variables and 50 sample points when the variables are 4.

[47]: Jones *et al.* (1998), 'Efficient global optimization of expensive black-box functions'

No more recent or rigorous study than the ones already presented was found in the literature that formally analyses this question. Thus, this thesis research has the goal to fill this gap. The quality of meta-models will be assessed when increasing the number of samples on analytical test functions. The range of points that will be used will consider the 10d rule and the conclusions drawn by Le Bihan. Also, the maximum budget expected for a project at Van Oossanen N.A. will mostly condition this decision.

3.3.2 Sampling methods

A second important question that requires an answer is how to spread points in the domain. A well-posed problem allows the user to explore the domain both globally, in search for possible areas of minimal resistance, and locally. Moreover, in particular, the query is what influences the choice of which sampling method to select. Two main factors are recognised:

- The meta-model that will use the points, and its settings;
- The nature of sample points, with particular care to the possible presence and the kind of errors.

As for the first point, the issue is postponed to Chapter 6 with the investigation of different meta-models and their requirements in terms of samples. About the other matter, computer experiments, like CFD simulations, as opposed to physical experiments that have a random error due to measurement inaccuracies, are deterministic (i.e., the same output is obtained each time the same input is given). The conclusion is that the primary source of error, causing "noise" in the result, is systematic and noise in the CFD evaluations required to produce the sampling points is inevitable.

It is calculated by several authors, including Wilson et al. (2001) [99], that noise in CFD simulations of hulls does not exceed 5%[5] . For this reason, as Sacks et al. (1989) [79] stated, points should be chosen to fill the design space rather than concentrate on the boundary. Jin et al. [44] confirmed in 2001 that a consensus among researchers is that experimental designs for deterministic computer analyses should be space-filling.

5: Data also confirmed by Van Oossanen N.A. for the simulations in situ

Four types of space filling sampling methods are relatively more often used in the literature [97]. These are orthogonal arrays [67] [34], various Latin Hypercube designs [60] [68], Hammersley or Halton sequences [32] [31], and uniform designs [21]. Uniform designs and

Hammersley and Halton sequences belong to a more general group called low discrepancy sequences [12].

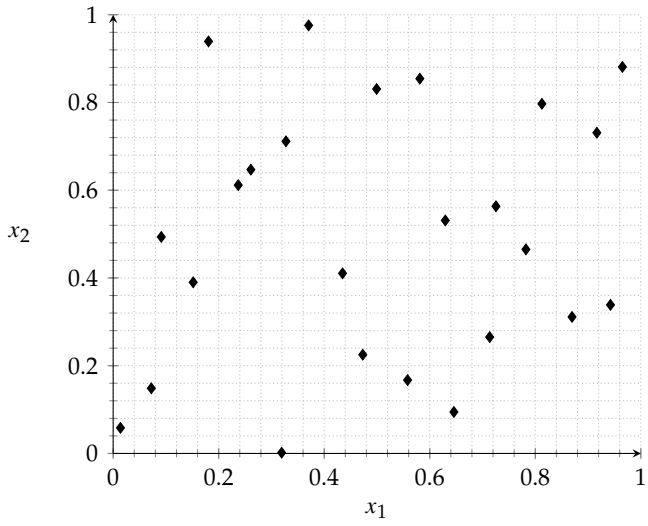


Figure 3.3: LHS sampling example in $\mathbb{R}^2$ with $p = 25$ sampling points

In the case of SBO applied to Naval Architecture, different authors in literature have chosen different space-filling sampling methods. For example, Guerrero et al. (2018) [28] and Raven et al. [72] [73] decided to use LHS method. However, the work of Simpson et al. (2001) [86] pointed out how the Hammersley sequence gives overall better results than LHS. At Van Oossanen N.A., instead, Halton sequence was chosen, because it does not create small clusters in some area of the design space, which could lead to a singular matrix in the case of Kriging and Radial Basis Function meta-models [16].

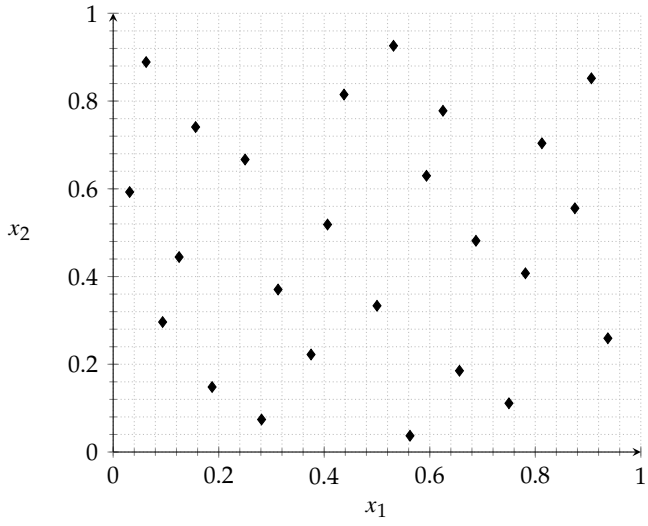


Figure 3.4: Halton sequence example in two dimensions with 25 sampling points

Given this literature review, the two most promising methods are LHS and Halton sequence. An example of these two methods for $n = 25$ sampling points is given in Figure 3.3 (LHS) and 3.4 (HALTON). Also, both are easily supported by DAKOTA toolkit. Thus, these two methods will be analysed and compared with the use of DAKOTA toolkit on several test functions.

3.3.3 Infill sampling criteria

The surrogate model $\widehat{f}$ is only an approximation of the true function f. It was constructed from an initial set of sampling points that were selected with a space-filling technique when the user did not know the function f itself. Jin et al. (2001) [44] also stated that the goodness of a fit obtained from an initial sampling plan solely is not sufficient to assess the accuracy of newly predicted points. Therefore, it is prudent to enhance the accuracy of the model using more points. This refinement step needs new algorithms able to intelligently select new *infill* points in addition to the initial sampling plan.

[44]: Jin *et al.* (2001), 'Comparative Studies of Metamodeling Techniques Under Multiple Modeling Criteria'

The first question that arises is how many points should be part of the infill plan. According to Forrester et al. (2008) [23] , the answer differs if the wish is to find a local or a global minimum. In the former case, only a few points should suffice for an infill plan, and the majority should be part of the sampling plan. However, the risk is to waste precious evaluations by selecting them upfront without regard to the response surface. On the other hand, according to Forrester, if the goal is the global minimum, most of the points should be positioned using an infill criterion rather than a sampling plan. Nonetheless, in this case, could the approximation based on a tiny sample be accurate enough and thus not misleading? In general, studies have shown that approximately between one-third and two-thirds of the total number of points available should be part of the infill plan [88].

[23]: Forrester *et al.* (2008), *Engineering Design Via Surrogate Modelling: A Practical Guide*

Exploration vs Exploitation

In general, the flowchart of the infill process is displayed in Figure 3.5, taken from [43] . It is shown how new sample points are selected based on specific infill criteria to update the surrogate model until the iterative process is terminated. The stopping criteria may be that convergence is reached, or that simulation budget is ended. Following several authors (Jones 2001 [46], Sobester et al. 2005 [88], Forrester et al. 2008 [23], Jiang et al. 2020 [43]) we recognise three possible criteria to select new infill points: a global exploration, local exploitation, and a method that balances the previous two techniques.

[43]: Jiang *et al.* (2020), 'Surrogate model-based engineering design and optimization'

Local exploitation gives a way to improve the accuracy in the region of the optimum predicted by the surrogate model. The most common criterion, in this case, is to minimise the response surface [43]: for each iteration, the surrogate model will predict the position of the best point. This point will be evaluated and given back to the surrogate for next iteration [46]. This kind of criteria has the advantage that it can be used with any kind of surrogate model, and the speed of convergence is fast [43]. Also, it is usually efficient for finding a local minimum, but it can easily miss a global minimum [46]. Especially for multi-modal functions, if the initial meta-model does not approximate the full function well, an infill strategy that can also search away from the current minimum and explore other regions is required [23].

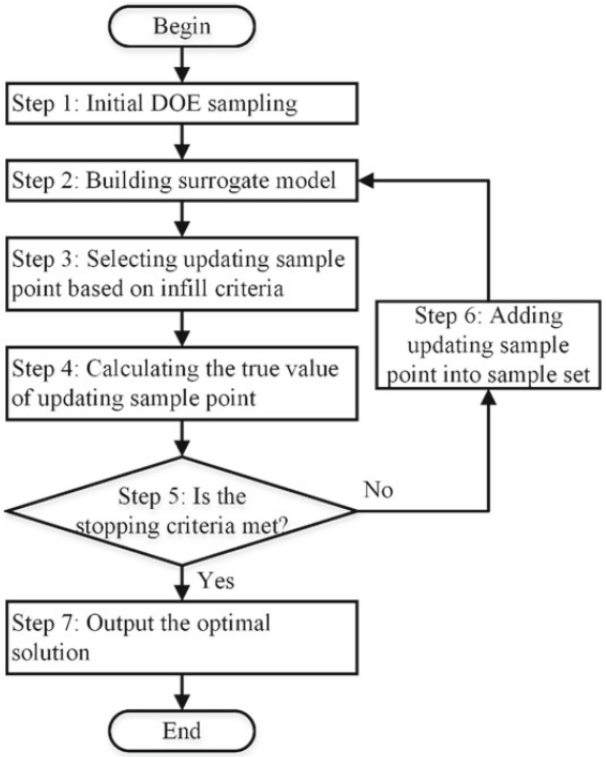


Figure 3.5: Flowchart of the infill part of the SBO algorithm. From Jiang et al. (2020) [43]

Global exploration adds points in areas of limited sampling where the uncertainty of the surrogate is high. This means increasing the chance of not missing any minimum previously not captured [23]. Criteria include a Mean Square Error (MSE) approach [23], where selected points are the ones with the largest estimation of prediction error, but it can be used only for Gaussian process-based surrogate model. Alternatively, for example, a Maximin Distance Approach [45], that selects points that maximise the minimum distance between any two sample points in the sample set. These and other methods have a good space exploration performance; however, the feeling is that sometimes the same output could be achieved by simply using a larger sampling plan. Also, it is not clear when should one stop adding points [23]. Moreover, the accuracy of the final solution cannot be guaranteed, and the overall number of calculation is large [43].

Balanced criteria. Instead of either exploiting or exploring the model, an infill criterion which balances both options can be used. Such criteria try to pursue a trade-off between being ideally very accurate near the global/local optimal location and acceptably rough elsewhere [39]. Two of the most used and promising algorithms are the Probability of Improvement (PoI) and Expected Improvement (EI) [46] [23] [43]. However, both methods necessarily require a Gaussian process-based meta-model, such as Kriging. These methods provide not only the predicted response value at any point of the domain space but also the corresponding prediction error associated. Typically, the prediction error is in the form of a standard deviation from the mean predicted value. This way, each iteration, a point can be added where the

improvement is statistically greater. These methods were found to be convergent, but sensitive to the choice of initial points and of convergence criteria. A deceptive sampling plan can cause the surrogate model to underestimate the true error and, as a result, converge prematurely or slowly [46].

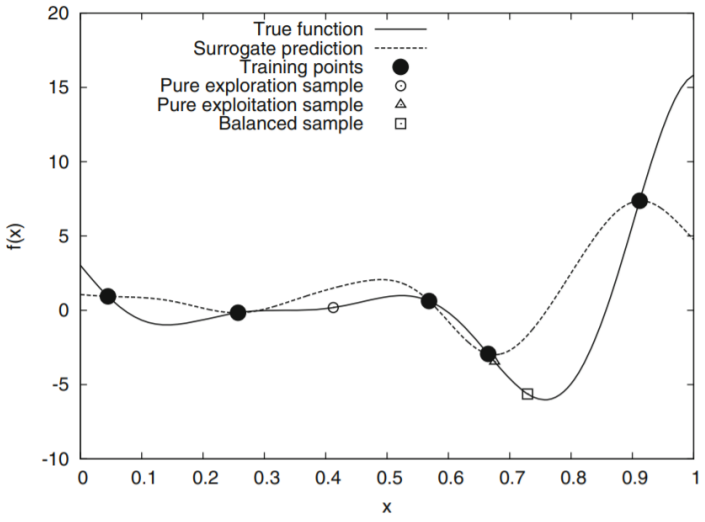


Figure 3.6: Exploitation vs exploration, 1D example. From Iuliano and Perez 2016 [40]

Figure 3.6 provides a simple example of adding a new training point by using, respectively, exploitation, exploration and balanced approaches. Given a set of training points (black circle points) evaluated on the exact function (solid black line), a surrogate model (dashed black line) is built: if a new sample has to be added, a simple exploitation approach would place it where the global minimum of the surrogate is detected, i.e. very close to one the training point (triangle point); a simple exploration approach, instead, would lead to sample where the maximum uncertainty in the model prediction is found, i.e. far from available training points (circle point); a balanced exploration/-exploitation approach combines the two aspects, thus providing a new sample which significantly improves the surrogate prediction [40] .

[40]: Iuliano *et al.* (2016), *Application of Surrogate-based Global Optimization to Aerodynamic Design*

In the case of SBO applied to Naval Architecture, not many authors have chosen to use an infill technique. For example, Scholcz et al. (2017) [83] used the local criteria of minimising the response surface. They conclude that the approach is promising, but mainly when performing a Multi-Objective Optimisation. Also, Guerrero et al. (2018) [28] demonstrated that the surrogate can be improved by adding new points to it. However, even though they acknowledged the possibility of using an Expected Improvement algorithm, the location of these new points was manually selected. Iuliano and Quagliarella in 2013 [41], for their foil shape optimisation, used the EI function. They concluded that it is found to be proper balancing between the need to exploit the approximated surface with the need to improve the approximation overall. At Van Oossanen N.A. the EI function is also used [16], but complaints have often arisen that the algorithm tends to choose points too close to each other. Furthermore, therefore the number of refinement points was always kept low. Thus, simulation time is wasted because from one CFD simulation to the next, no real new information is added to the meta-model.

To conclude, this thesis will explore different possibilities. However, only a proper balance of both exploration and exploitation ensures a higher chance of convergence towards the global minimum of the function to be approximated [39]. To achieve this balance, the Expected Improvement function will be further on analysed as it looks like the best candidate for this work. It is also implemented in DAKOTA toolkit. However, this technique is no longer applicable when meta-modelling techniques other than Kriging are used [45]. In general, in the literature of computer experiments, sequential sampling approaches with adaptation are mainly developed for the Kriging [79] method. Therefore, also a combination of minimisation of response surface and Maximin Distance Approach will be further analysed, both also supported in DAKOTA. This way, we ensure a solution usable with any meta-model.

3.4 Surrogate models

The core problem is attempting to learn a mapping $y = f(\boldsymbol{x})$ that lives in a black box. Therefore, the physics that converts the vector x into a scalar output y are unknown. This black box could either be a physical or computer experiment, as in this case. The generic solution method is to gather the output values $y^{(1)}, y^{(2)}, \cdots, y^{(n)}$ that originate from a set of inputs $\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \cdots, \boldsymbol{x}^{(n)}$ and find a best guess $\widehat{f}(\boldsymbol{x})$ for the black box mapping f, based on these known observations. [23]

3.4.1 General overview

[56]: Mason (2010), 'Stochastic Optimisation of America's Cup Class Yachts'

One of the greatest issues when dealing with meta-models is understanding which factors influence the choice of which one to use for a particular optimisation task. According to Mason (2010) [56] some of the aspects are:

- Quantity of data available: the choice may differ if only a small number of data points are available, for example when the number of points is strictly linked to the computational cost, for instance, because of CFD simulations;
- Dimensionality of the solution space and region of validity: local, mid-range, and global optimisation [8];
- Complexity of the solution surface: not every model can capture a highly non-linear behaviour, in comparison with a smoother one;
- Degree of noise associated with the data: not every model is equally influenced by noise, nor every model deals with it the same way;
- Ease of use: a surrogate model is successfully implemented, in the case of this thesis, but also in general within a company, when it is easy to test and validate it. Also, it needs to be efficiently tuned to fit various situations, and when information can be derived from the results quality and prediction accuracy.

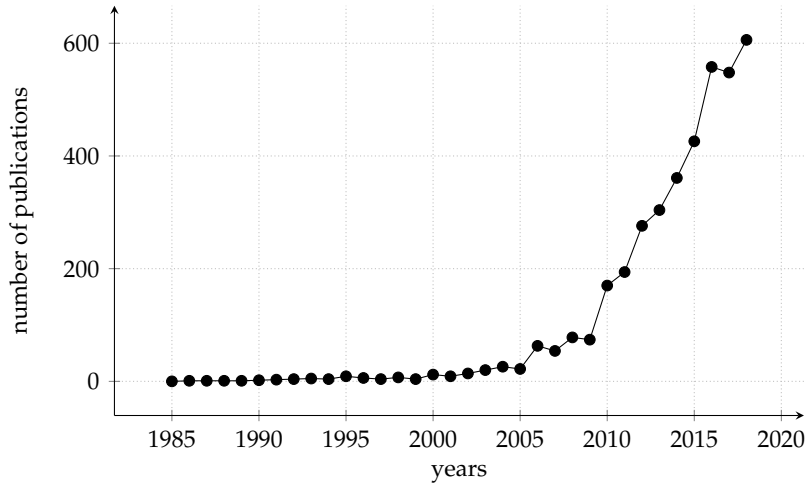


Figure 3.7: Approximate number of publications from 1985 to 2018 when searching the words "surrogate based optimisation" in Google Scholar

The use of the surrogate model technique raised during the years, as shown by the increasing number of publications in Figure 3.7. However, many alternative meta-model formulations exist and may be suitable for the case of this thesis, in particular the list include Polynomial Regression (PR), Moving-Least Square (MLS), Multivariate Adaptive Regression Splines (MARS), Kriging (KG) [28] [73], Radial Basis Functions (RBF), Artificial Neural Networks (ANN) [56], and Support Vector Machines (SVM). All are widely used in optimisation routines, but no method has shown to be overwhelmingly superior across a range of approximation tasks.

Table 3.4: Summary of best methods. Analysis performed by Jin et al. (2001) [44]

	High-order Nonlinear	**Low-order Nonlinear**	**Overall**
Large scale	RBF	Kriging	RBF
Small scale	RBF	PR	RBF
Overall	RBF	PR	RBF

Several authors have performed comparative evaluations. Jin et al. (2001) [44] systematically compared four popular meta-modelling techniques — Polynomial Regression, Multivariate Adaptive Regression Splines, Radial Basis Functions, and Kriging — based on multiple performance criteria. The author used fourteen test settings representing different classes of problems and different levels of linearity (top part of Table 3.4). Also, different tests were performed on the functions both with limited or large sample sets (left part of Table 3.4). The models were evaluated in accuracy and robustness, performance for different types of problems and sample sizes, behaviour in case of noise, efficiency, transparency, and simplicity. The summarised results are visible in Table 3.4, where overall, the best performance was observed in RBF. It resulted in being the most dependable method in most situations in terms of accuracy and robustness, also with a limited sample

[44]: Jin *et al.* (2001), 'Comparative Studies of Metamodeling Techniques Under Multiple Modeling Criteria'

set.

Table 3.5: Recommendations for model choice and use. Analysis performed by Simpson et al. (2001) [86]

Model choice	Characteristics/Appropriate uses
Neural Networks	– Good for highly nonlinear or very large problems (~ 10000 parameters) – Best suited for deterministic applications – High computational expense (often > 10000 training data points); best for repeated applications
Kriging	– Extremely flexible but complex – Well suited for deterministic applications – Can handle applications with < 50 factors – Limited support is currently available for implementation

[86]: Simpson *et al.* (2001), 'Metamodels for Computer-based Engineering Design: Survey and recommendations'

Simpson et al. (2001) [86] reviewed several meta-models, including Artificial Neural Network and Kriging, surveying their existing application in engineering design. The recommendations they gave are summed in Table 3.5. In the opinion of Simpson et al., Neural Networks are suited to deterministic applications which require repeated use with a high number of training data points. In contrast, Kriging is an interpolation method extremely flexible and can work with fewer sample points; however, it is susceptible to noise.

Wang and Shan (2007) [97] presented the surrogate models from a practitioner's perspective, and Viana et al. (2014) [96] investigated how far meta-modelling techniques have evolved in the last two decades. The conclusion was that the future direction is to use multiple surrogates to better adapt to every possible situation. Readers interested in other model comparison works, also specifically for naval architecture applications, are referred to other articles on the topic [14] [56] [71] [82].

Table 3.6: Comparison between surrogate models. Analysis performed by Q. Delivré (2014)[16].

Criteria	PR	MLS	MARS	KG	RBF	ANN	SVR
Lowly non linear	1	1	1	0	0	1	1
Mediumly non linear	-1	1	1	1	1	1	1
Highly non linear	-1	0	0	1	1	1	1
Low dimension	1	1	0	1	0	0	1
Medium dimension	0	0	1	1	1	1	1
High dimension	-1	-1	1	1	1	1	1
Tuning time	1	1	1	-1	0	-1	-1
Prediction time	1	1	1	-1	0	1	0
Validation time	1	1	0	0	0	-1	-1
No prior knowledge	-1	-1	-1	1	1	-1	1
Use	1	0	-1	1	0	0	0
Flexibility	-1	-1	-1	1	1	1	1
Multifidelity use	-1	-1	-1	1	0	1	0
Total	0	2	2	7	6	5	6

Also within Van Oossanen N.A. effort has been put in analysing the best surrogate model for applications typically encountered in the company, such as bulb shape or Hull Vane optimisations. In particular,

Delivré [16] investigated seven possible models for thirteen criteria assigning a value of -1, 0, and 1 to respectively a bad, average, and good performance. The results are shown in Table 3.6, where the less intuitive criteria are described from Delivré as:

[16]: Delivré (2014), 'Optimisation de carène par modèle de substitution'

Validation time means if the methods offer (or not) a statistical criterion to characterise the accuracy, and if not, an additional point has to be sampled to evaluate the deviation between the prediction and the sampling point;

No prior knowledge means that the user does not need to be experienced, or to have prior knowledge about the function behaviour;

Use means that the literature is abundant (or not) on these methods; so that results are proven, the library in programming code is well supplied, and some improvements are proposed;

Flexibility means it can adapt to various problems, it is partially redundant with the "non-linear" and "dimension" blocks, it is a kind of bonus point since it is an important aspect;

Multi-fidelity use means that the literature is abundant (or not) on Multi-fidelity, results are then proven, code library is well supplied, and some improvements are proposed.

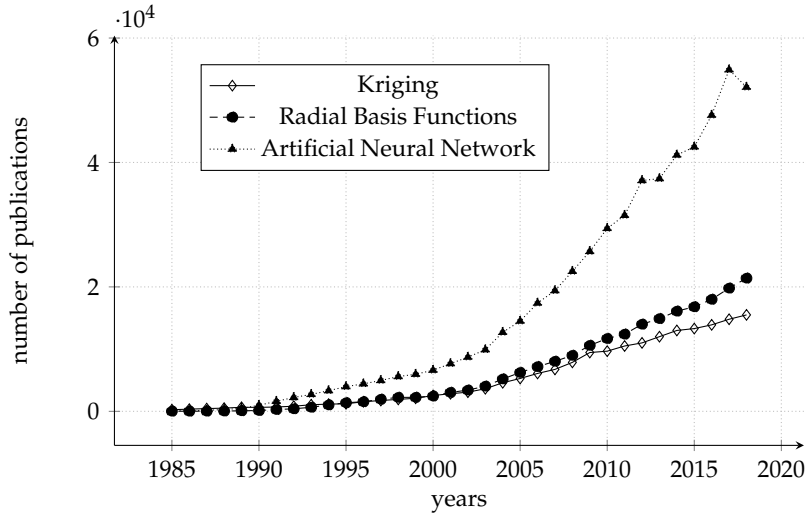


Figure 3.8: Approximate number of publications from 1985 to 2018 when searching the different surrogate models in Google Scholar

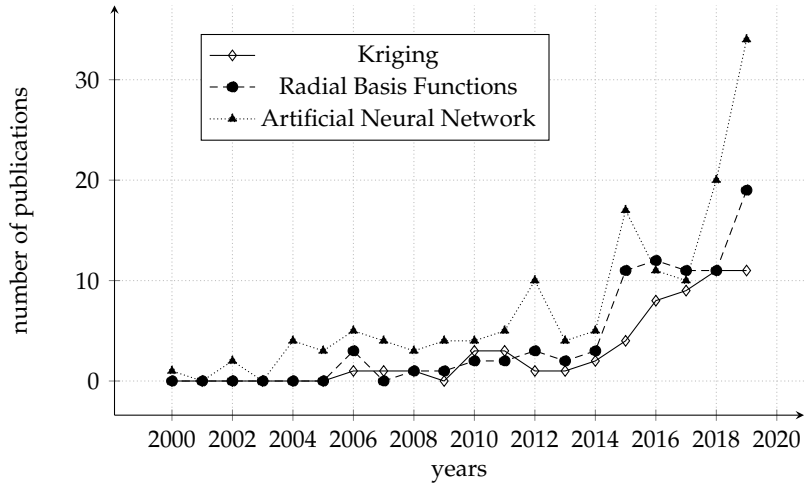


Figure 3.9: Approximate number of publications from 1985 to 2018 when searching the different surrogate models and "ship resistance" in Google Scholar

Delivre's work was based solely on literature as well as mostly

subjective. As a consequence of it, the Kriging method was chosen because the sum of all the points was higher than for the other models. Thus, a more accurate comparison is required and thus performed in the context of this thesis. For the sake of simplicity and pragmatism, not all seven models will be analysed. The three methods that from this literature analysis demonstrated to have better performances for our purposes and are also easily supported by DAKOTA are Artificial Neural Networks (ANN), Radial Basis Functions (RBF), and Kriging (KG).

A demonstration that the three selected method can indeed be suitable, is given from Figures 3.8 and 3.9. They show the number of publications from 1985 to 2018 when searching the different surrogate models (upper) and the different models in combination with the words "ship resistance" (lower), in Google Scholar. Note that these results may slightly vary due to the update of the Google database. Both figures show a steady growth of publications for all three methods, especially with regards to ANN.

3.4.2 Assessing model accuracy

A second important question about meta-models to answer is how to assess meta-model accuracy and what to look for when doing so. When in the context of this thesis, the different models will be analysed and confronted, their performance will be measured, taking into account the following aspects [44] :

[44]: Jin *et al.* (2001), 'Comparative Studies of Metamodeling Techniques Under Multiple Modeling Criteria'

- Accuracy – the capability of predicting the system response over the design space of interest. In particular, sensitiveness to linearity and dimension of data;
- Robustness and flexibility – the capability of achieving good accuracy for different problem types and sample sizes;
- Efficiency – the computational effort required for constructing the meta-model and for predicting the response for a set of new points by meta-models.
- Transparency – the capability of illustrating explicit relationships between input variables and responses;
- Conceptual Simplicity – ease of implementation. Simple methods should require minimum user input and tuning time and be easily adapted to each problem.

The quality of a meta-model has a profound effect on the computational cost and convergence characteristics of the Surrogate Based Optimisation [102]. A surrogate model that was built on a limited number of sample points will inescapably have significant prediction uncertainty. Applying such unreliable surrogate models in design and optimisation may lead to misleading predictions or optimal solutions located in unfeasible regions [70]. Therefore, verifying the accuracy of a meta-model before working with it can ensure the reliability of the entire design [43].

The applications of error metrics in the design optimisation can be generally classified into four cases [1] , of which only the first two are of our interest:

[1]: Acar (2015), 'Effect of error metrics on optimum weight factor selection for ensemble of metamodels'

1. identifying regions with relatively high uncertainty in the input domain to determine promising areas for model refinement;
2. obtaining an overall assessment of the constructed surrogate model to be used for prediction, uncertainty quantification or optimisation;
3. for an ensemble of surrogate models, determining the optimal weight factors for the individual surrogate models;
4. choosing the most appropriate model among the alternatives when multiple surrogate models are available.

When the designers must decide whether to accept or reject the constructed surrogate model, the overall accuracy of the surrogate model needs to be known, and the metric value should be as close to the true error as possible since the criterion for accepting a model is often set in reference to the true error. There are many alternatives available when choosing an error metric to assess the accuracy of a surrogate model. Among others, Meckesheimer et al. (2002) [61], Sacks et al. (1989) [79], and DAKOTA's User manual [5] propose a cross-validation method with the use of Root Mean Square Errors.

Metrics

To compare different meta-models, or assess their accuracy during an SBO, DAKOTA comes with the ability to compute diagnostic metrics based on [5] :

[5]: Adams *et al.* (2019), 'Dakota, A Multilevel Parallel Object-Oriented Framework for Design Optimization, Parameter Estimation, Uncertainty Quantification, and Sensitivity Analysis: Version 6.10 User's manual'

- Simple prediction error for the training data;
- prediction error estimated by cross-validation (iteratively omitting subsets of the training data);

All diagnostics are based on differences between $o(x_i)$ the observed value, and $p(x_i)$, the surrogate model prediction for training (or omitted or challenge) data point x_i. In the simple error metric case, the points x_i are those used to train the model, for cross-validation they are points selectively omitted from the build, and for challenge data, they are extra points provided by the user. The metrics include:

Root Mean Square Error (RMSE) can be regarded as the average vertical distance of the actual observations from the fit line. In this method, points with larger errors tend to be assigned higher weights by penalising the variance. The formula for the RMSE when evaluated at a set of test points n is as follows:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\Big(o(x_i) - p(x_i)\Big)^2} \tag{3.6}$$

Mathematically, the RMSE is also a global error metric, and a smaller value represents a higher overall [97] accuracy [103] [84].

Maximum Absolute Error (MaxAE) is a local [97] error metric that measures the maximum approximation error of the surrogate model [84]. The formula for the MaxAE for a set of test points can be expressed as follows:

$$\text{MaxAE} = \max_i \left| o(x_i) - p(x_i) \right| \qquad i = 1, \cdots, n \tag{3.7}$$

Cross-Validation Error

Cross-Validation (CV) was originally proposed in the 1930s [52], and it was further developed and refined in the 1970s by Stone [90] . In the CV method, the above metrics can be computed via a cross-validation process. The class of k-fold cross-validation metrics is used to predict how well a model might generalise to unseen data. The training data is randomly divided into k partitions. Then k models are computed, each excluding the corresponding k^{th} partition of the data. Each model is evaluated at the points that were excluded in its generation, and RMSE or MaxAE are computed for the held-out data. In the end, all metrics are averaged over the number of evaluations. A special case, when k is equal to the number of data points, is known as *leave-one-out* (LOO) cross-validation or PRediction Error Sum of Squares (PRESS) [5]. It is worth mentioning that LOO will brake the space-filling properties of sampling plans; however, no alternative is given by DAKOTA toolkit.

[90]: Stone (1974), 'Cross-validatory choice and assessment of statistical predictions'

CPU time

The time consumed in the process is a measure of efficiency. The less time the regression process spends, the more efficient a meta-model is [103] [66]. However, it is already predicted that this will not be of major importance to this thesis. It is not because CPU time for constructing a surrogate is in the order of seconds, while CFD simulations are in order of hours.

[103]: Zhao *et al.* (2010), 'A comparative study of metamodeling methods considering sample quality merits'

3.4.3 Visualisation

As already explained, a vital part of the Surrogate Based Optimisation is the possibility for the user to visualise the data in such a way that it is easy to extract results and detect anomalies. Since the routine is expected to be used for commercial projects, an informed choice on the final optimised geometry is crucial.

There are two methods that in literature stand out for completeness and clarity. The first is found in the work of Guerrero et al. [28] [26] and an example is given in Figure 3.10. This scatter matrix, in one single illustration, shows the correlation information, the data spreading using histograms and scatterplots, and regression investigation of the responses of the quantity of interest. For this example, a function with six design variables (DV1, DV2, ..., DV6 on the left and bottom) was used, and the objective is called a Quantity of Interest (QoI).

[28]: Guerrero *et al.* (2018), 'Surrogate-Based Optimization Using an Open-Source Framework: The Bulbous Bow Shape Optimization Case'

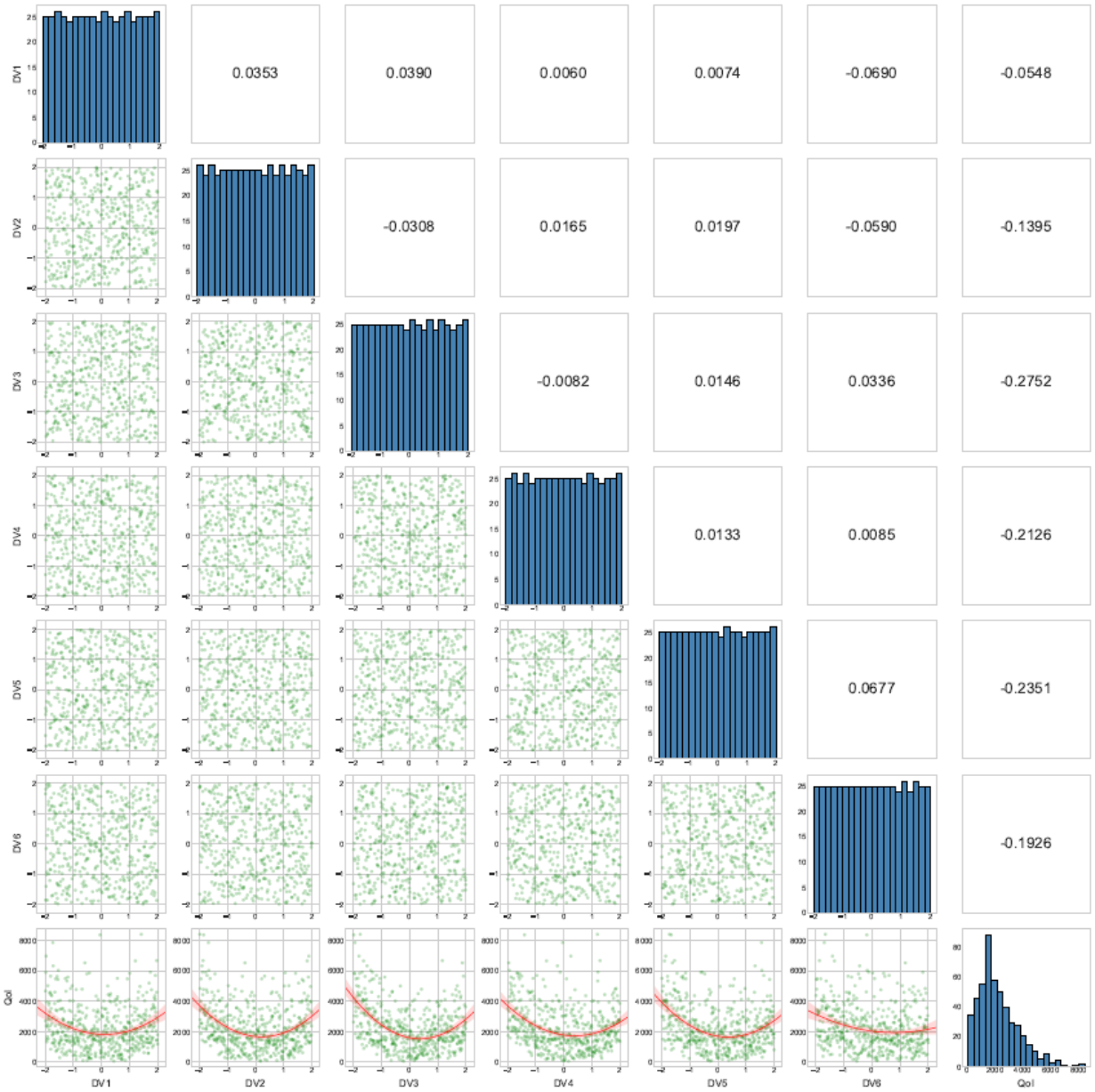


Figure 3.10: Scatterplot matrix of the high-dimensional Rosenbrock function space exploration study ($d = 6$). The Spearman correlation is shown in the upper triangular part of the matrix. In the diagonal of the matrix, the histograms showing the data distribution are displayed. In the lower triangular part of the matrix, the data distribution is shown using scatterplots. In the last row of the matrix plot, the response of the quantity of interest (QoI) as a function of the design variables is illustrated, together with a quadratic regression model. From Guerrero et al. (2018) [28]

By conducting a quick inspection of the figure, we can see from the histograms and the scatterplot of the experiments (lower triangular part of the matrix) that the distribution of the data in the design space is uniform. If we observe regions in the design space that remain unexplored, we can simply add new training points to cover those areas. In the case of outliers (anomalies), we can remove them from the dataset with no significant inconvenience. However, we should be aware that outliers are telling us something, so it is a good idea to investigate the cause and effect of the outliers. In the upper triangular part of the plot, the correlation information is shown (Spearman correlation, in this case). This information tells us how correlated or uncorrelated

the data are. For example, by looking at the last row of the plot that shows the response of the QoI, if we see a strong correlation between two variables, it is clear that this variable cannot be excluded from the study. The opposite is also exact; that is, uncorrelated variables can be excluded; therefore, the complexity of the problem is reduced. Additionally, the last row of the scatter matrix plot also shows the surrogate model (a quadratic model, in this case) and its trends.

The other visualisation tool that is going to be used in this thesis was developed within Van Oossanen N.A.. An example is given in Figure 3.11, where we can see some plots equal to the number of variables (four, in this case: chord, span, x position and z position of a Hull Vane®). For each plot, we can see resistance over one of the variables while keeping fixed the values of the other variables to the ones that were found to be optimal. All sampling points (the numbers) are depicted, as well as the refinement ones (red) and the optimum found (in green). The lines represent the mean of the Kriging surrogate model (black) and a confidence interval of the prediction (red). If needed, more information can also be added to this plot, such as the bare hull resistance (blue line).

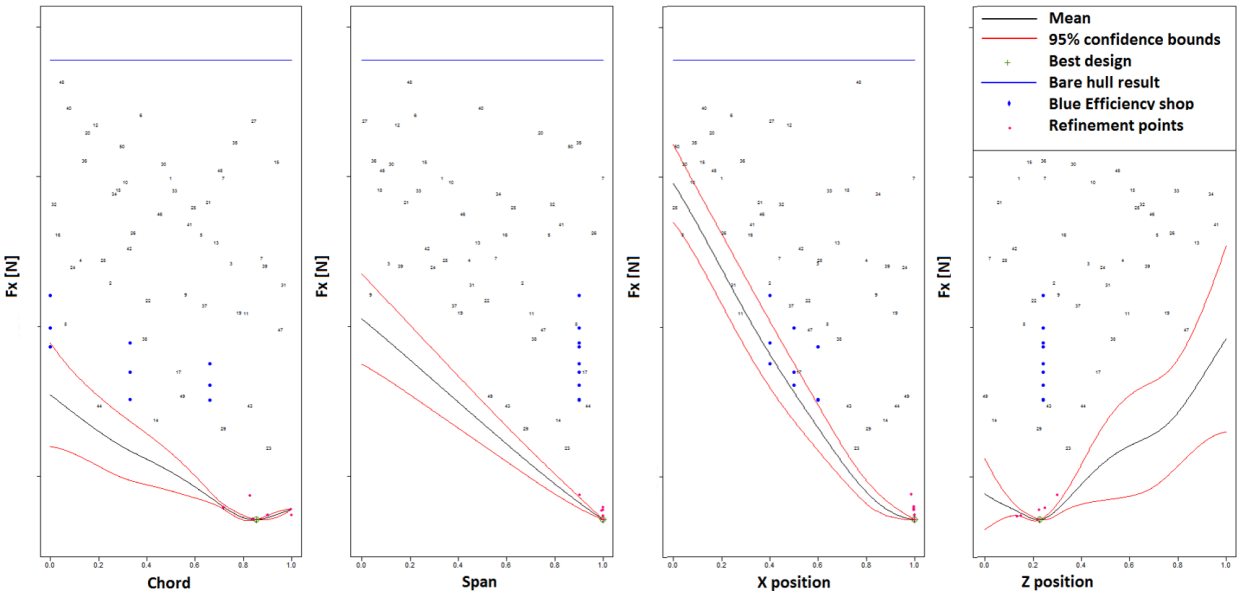


Figure 3.11: Surrogate Based Optimisation visualisation tool. Courtesy of Van Oossanen N.A.

3.5 Conclusive remarks

SBO consists of several interconnected steps, and each increase complexity and uncertainty to the overall process. During this literature review phase, we recognised the need to investigate SBO methods and especially the effects of different meta-models when it comes to applying them to a hull shape optimisation typical at Van Oossanen N.A.. Moreover, no standard rules were ever proposed for how to proceed. Therefore the thesis is trying to list guidelines to follow when building up an SBO routine in this specific marine field. The central research part is devoted to the study and comparison of possible methods, their

advantages and drawbacks, and their performance, for every sub-topic identified.

Firstly, an overview of what is optimisation is given. The number of possible algorithms that can be used is vast, but can be lowered down analysing the nature of design variables, of objective functions, and constraints. Also, time limit and availability of the algorithm implementation can help in the decision process. For this application, the most promising solution is a combination of a global gradient-free population-based algorithm (GA) and a local one.

Table 3.7: Summary of conclusions drawn from literature review

Item	Most promising solution
Optimisation	Combination of global and local algorithms – Global: GA – Local: PS or QN
Sampling	Sample size – investigation required on tests to select a number – keep in mind: (a) maximum number of simulations allowed at Van Oossanen N.A.. (b) $10d$ rule [47] and Le Bihan's conclusions [54]
	Sample method – LHS or Halton sequence
Infill	Balanced required between exploration and exploitation – if KG is used: EI looks promising. To investigate. – if KG is not used: combination of two methods
	Convergence criteria – investigation required for chosen method
	Percentage of initial points over the total – investigation is necessary – keep in mind: Sobester's recommendations to stay between 35% and 60% [88]
Surrogate	Most promising: KG, ANN, RBF – investigation is necessary on test functions – keep in mind: (a) KG is the most used and supports EI. (b) noise is inevitable and affects differently the models

Secondly, the initial sampling choice is analysed. The main concerns recognised for this sub-topic are the sample size and the sample nature. Nature is both by the distribution of points over the design space (sampling method) and the presence of error in the experiments that generate the points. For the first issue, the decision is to investigate the effect of increasing sampling points on the quality of surrogate models with the use of test functions. About the second, as noise is inevitable in CFD simulation, the best possible methods seem to be the Halton sequence and Latin Hypercube.

Then, the infill sampling criteria (how to improve the accuracy of the surrogate adding points to the initial sampling plan) is described.

The biggest issue is how to obtain a balance between exploration and exploitation of the design space. For this application, the right solution could be the Expected Improvement function.

Finally, a generic discussion and overview of meta-models are given. When choosing which one to use, some important aspects are the quantity of data available, the dimensionality and the complexity of the solution surface, and the ease of use. After careful investigation, it looks like the three best models to test that are also available in DAKOTA are Artificial Neural Network, Radial Basis Functions, and Kriging. To properly test these methods, accuracy, robustness, efficiency, and transparency must be taken into consideration. Few techniques were found thriving in the literature to estimate accuracy: prediction error concerning training data (RMSE, MaxAE), and prediction error estimated by Cross-Validation. In the end, visualisation tools are recognised as extremely important for the user to extract results and detect anomalies. Scatter plots and histograms are among the most used, because of their ease of use and generation.

SURROGATE BASED OPTIMISATION: MAIN RESEARCH

Optimisation 4

In Section 2.1.1, when SBO workflow was presented along with its issues, the main problem about optimisation has risen: what method is best for the kind of surfaces that are most commonly faced? Moreover, what influences that choice? We tried to give an overview of the answers that other authors in literature gave to these questions in Section 3.2. We saw that the number of possible algorithms that can be used is vast, but it can be lowered down analysing the nature of design variables, of objective functions, and constraints. As a reminder, Table 4.1 summarises the typical features that affect the choice of optimisation algorithms typically encountered during this thesis at Van Oossanen N.A.

Table 4.1: Summary of optimisation characteristics typical at Van Oossanen N.A.

Item	Characteristic
Design variables	From 2 to 4 (they represent geometrical features)
Objective function	– Single objective (Hull total resistance) – Possibility of extending to MOO or MDO in the future
	Surrogate surface: Cheap to compute, continuous, and always positive
	"Black-box" function: No assumptions can be made on convexity, smoothness, (non-)linearity, (multi-) modality
Constraints	Simple bound (usually $0 \leq x_n \leq 1$)

Also, time limit and availability of the algorithm implementation can help in the decision process. In particular, for this application, the most promising solution that was found (see Table 3.7) is a combination of a global gradient-free population-based algorithm and a local one. As for the former, Genetic Algorithm appeared to be the most widely used and the most suitable. While for the latter, the gradient-free Pattern Search algorithm and the gradient-based Quasi-Newton algorithm are the two best options available in DAKOTA.

4.1 Global search: Genetic Algorithm

Genetic Algorithm (GA), also called evolutionary strategy, uses "survival-of-the-fittest" to weed out poor solutions and breed new solutions to an optimisation problem [7]. It is based on the abstraction of Darwin's evolution of biological systems, and J. Holland and his collaborators

pioneered it between the 1960s and 1970s [36]. According to this theory, an individual (a geometry) with favourable genetic characteristics (design variables) is most likely to produce better offsprings (lower resistance). Selecting them as parent increases the probability that the individuals of the next generation will perform better than the previous one. Information within a point is encoded as chromosomes, and biological processes such as crossover and mutation are used to create new individuals.

A GA based search can be viewed as a trade-off between the maintenance of sufficient diversity in the population to permit the coverage of the entire solution space (exploration), and the need to converge on the optimal solution within an acceptable time (exploitation) [19].

Genetic algorithms have two main advantages over other algorithms: they can deal with complex problems and parallelism. Whether the objective function is stationary or transient, linear or non-linear, continuous or discontinuous, it can be dealt with by genetic algorithms [50].

4.1.1 Workflow and parameters

The basic steps of a GA are depicted in Figure 4.1. Each step depends on several variables or algorithms that can be adjusted to improve the performance of the GA or to manipulate the degree of exploration or exploitation that the GA exhibits [24]. For each of them, many different possibilities can be used, but here only the main idea of how they work is given. For more information readers are referred to manuals on the topic [7] [24] [29].

[24]: Goldberg (1988), 'Genetic Algorithms in Search Optimization and Machine Learning'

Initial population The first step is the selection of a random initial population. Its main characteristics are type and size;

Fitness Then, the algorithm has to select suitable parents from the initial population and place them into a mating pool. These individuals are further used for mating and generating new offspring. Since the characters of these individuals are passed to the next generation, only the individuals who have desirable properties are selected. This transition is achieved by assigning fitness values to each individual to evaluate its quality;

Crossover and Mutation Given two parent individuals, crossover and mutation aim at creating an offspring individual that acquires attributes from the parents. Crossover is used to combine the desirable characters of two different parents. It is an analogous process to the recombination that occurs in sexual reproduction within a single chromosome (haploid) organisms. And mutation is used to maintain genetic diversity by randomly flipping bits of an individual chromosome;

Replacement A new population is defined after all new individuals have been evaluated. Replacement, which controls how previous populations and newly generated individuals are combined to create a new population, is employed;

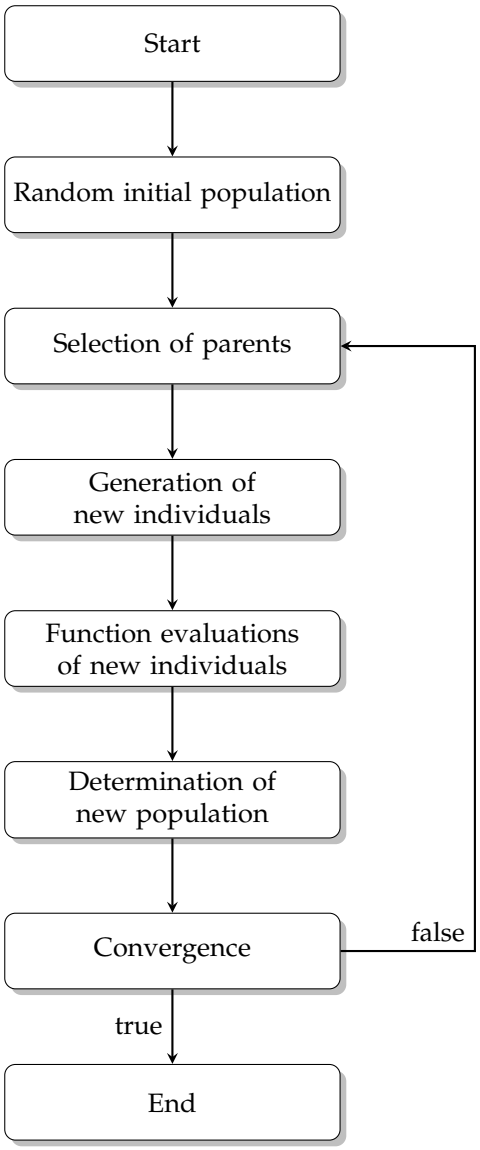


Figure 4.1: Genetic Algorithm workflow

Convergence and stopping criteria Finally, the algorithm returns to the selection of parents and continues until convergence criteria are satisfied, or iteration limits are exceeded.

4.1.2 GA in DAKOTA

Genetic algorithms available in Dakota include `coliny_ea` and `moga` methods. The former can return as output more than one best solution, in order of lowest response function value while the latter is specifically designed for multi-objective problems. Both of them follow the structure seen in Figure 4.1, and the user can modify the parameters described in Section 4.1.1.

In the context of this thesis, `coliny_ea` is going to be used because the focus is not on a MOO, and because it allows us to define more than one possible starting point for the local search. This additional feature allows GA to pass to the local algorithm more than one starting point, and thus chances of not missing any local solution are enhanced.

Table 4.2 gives a summary of which parameters are going to be used in the validation tests in Chapter 8 and Chapter 9. The methods and numbers were selected with two main criteria: or they are default and recommended options available in DAKOTA, or they are commonly recognised among researcher as a guarantee of the best solution. Further investigations could be performed in future work; however, for the scope of this thesis, the accuracy obtained with these values is judged to be enough.

Table 4.2: Summary of parameters used for GA in DAKOTA

Item	Characteristic
Initial population	Size: $50d$ [a]
	Type: random (with eventual possibility to read the points from a file)
Fitness	Linear scaling of probability of parent selection based on the rank order of each individual's objective function within the population
Crossover	Size: 0.8 (DAKOTA default value) [b]
Mutation	Size: 0.01 [c]
	Type: mutation offset uses a normal distribution
Replacement	Size: 1 (DAKOTA default)
	Type: creates a new population using the `replacement_size` best individuals from the current population, and `population_size - replacement_size` individuals randomly selected from the newly generated individuals.
Convergence	Stopping criterion: if change in the objective function between successive iterations divided by the previous objective function is less than $1.E-4$.
	Maximum number of function evaluations: 1000 (DAKOTA default)

[a] d: dimension of problem, number of variables

[b] If 1, then the offspring will tend to look more like the first parent. If 0, then the offspring will tend to look more like the second parent. A value of 0.5, provides equal input from each parent [7].

[c] Recommendations for the probability of mutation $p \in [0,1]$ for each individual bit have been determined empirically by several researchers, including de Jong (1975) and Schaffer et al. (1989), and are typically in the range 0.005 – 0.01.

4.2 Local search: Quasi Newton

One of the most efficient and robust local trajectory gradient-based method is the Quasi Newton Broyden Fletcher Gordfarb Shanno (BFGS) algorithm [87] . It is an algorithm suitable to solve non-linear unconstrained optimisation problems. Furthermore, its basic idea is to seek a stationary point of a (preferably twice continuously differentiable) function. A necessary condition to find the minimum is that the gradient of the function is zero. In this method, the Hessian matrix

[87]: Sobester *et al.* (2014), 'Aircraft aerodynamic design: geometry and optimization'

is not computed, but it is approximated by analysing the successive gradient evaluations.

From an initial point x_0, its evaluation $f(x_0)$, its gradient $g_0 = \nabla f(x_0)$, and an approximation of the Hessian matrix $\boldsymbol{B}_0$, at each iteration k the algorithm performs the following steps [11]:

1. Calculate a search direction d_k by solving

$$\boldsymbol{B}_k \, d_k = -g_k \tag{4.1}$$

2. Perform a one direction optimisation (line search) to find an appropriate step size λ_k in the same direction found in the first step.
3. Update $x_{k+1} = x_k + \lambda_k d_k$.
4. Evaluate $g_{k+1} = \nabla f(x_k + \lambda_k d_k)$ and decide whether to stop. If not continue with next step.
5. Update Hessian approximation by setting $s_k = x_{k+1} - x_k$ and $y_k = g_{k+1} - g_k$ and calculating, before going back to step one:

$$\boldsymbol{B}_{k+1} = \boldsymbol{B}_k - \frac{\boldsymbol{B}_k s_k s_K^T \boldsymbol{B}_k}{s_k^T \boldsymbol{B}_k s_k} + \frac{y_k y_k^T}{s_k^T y_k} \tag{4.2}$$

This description hides many details of the method, for example, the calculations in the line search. Nevertheless, it is sufficient to indicate the essential characteristics of the method. For a more detailed description of the BFGS algorithm, see, for example, Dennis and Schnabel (1983) [17].

In general, the gradient information is not just useful for the algorithm to converge, but also for the user to identify what is called a robust solution. Practically, the optimised design that works well only in a given environment and performs poorly when the conditions fluctuate slightly is likely to be less desirable than one whose peak performance is lower, yet performs well in a variety of conditions. The second design is called robust. Moreover, the surface over which we are optimising is only approximated. Thus, we would often prefer a solution for which a small variation in parameters still ensures the Naval Architect to be in a resistance reduction area safely. If this does not happen, only a slight change in geometry due to other constraints or to construction uncertainties will lose the advantage given by the SBO. Nonetheless, this analysis can be done analysing the surface gradients visually without the explicit use of a gradient-based optimiser.

4.2.1 QN in DAKOTA

DAKOTA supports the BFGS method and allows the user to select the desired maximum step that can be taken when computing a change in the current design point. Also, stopping criteria can be set. It can both be a limit on the maximum number of evaluations or a convergence tolerance for the objective function or its gradient. The default DAKOTA setting sets the limit at $g_{k+1} < 1.E-4$ to indicate the reach of a stationary point.

4.3 Local search: Pattern Search

Pattern Search (PS) methods can be applied to non-linear optimisation problems when the information of gradients is missing or difficult to obtain. These methods are best suited for efficient navigation to a local minimum in the vicinity of the initial point. The selection of a suitable initial point has a great part in the success of the method.

PS algorithms generally walk through the domain according to a defined stencil of search directions. Some algorithms however have the possibility to control of how the search pattern is adapted. The number of trial points at each iteration grows linearly with the number of variables. For a n-dimensional problem, each iteration of the algorithm explores $2n$ directions. Of course, the number of iterations required to approach a local solution typically also increases with n. Given the function f, a starting point x_0, an initial step length $\delta_0 > 0$, a stopping tolerance $\epsilon_{\text{stop}} > 0$, and a solution target f_{target}, at each iteration k the algorithm will [7]:

1. Evaluate $f(t)$ for some $t \in P_k = \{x_k \pm \delta_k e_i, \text{ with } i = 1, 2, \cdots, n\}$ with e_i the standard basis for a n-dimensional space.
2. If $f(t) < f(x_k)$ than set $x_{k+1} = t$ and keep the step length ($\delta_{k+1} = \delta_k$). Otherwise, if $f(t) > f(x_k)$ than keep the starting point ($x_{k+1} = x_k$) and set $\delta_{k+1} = c\delta_k$ with c a contract factor often equal to 0.5.
3. Iterate point 1 and 2 until or $\delta_{k+1} < \epsilon_{\text{stop}}$, or maximum number of iterations is reached, or $f(x_{k+1}) < f_{\text{target}}$.

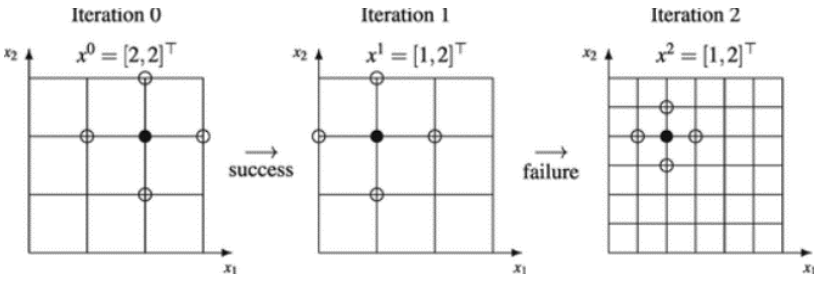


Figure 4.2: A successful and an unsuccessful iteration of a PS algorithm. From Audet and Hare (2017) [7].

In point 2 of the algorithm, if the improvement is found (the first case described), the iteration is called successful. Else, an unsuccessful iteration is one where x_k is shown to be an optimiser of f for the discrete set P_k, called the poll set. Figure 4.2 illustrates three iterations of a PS algorithm on a problem with $n = 2$ variables. The initial point is $x_0 = [2, 2]^T$ and is represented by the dark circle. The initial step length parameter δ_0 is chosen to be equal to one.

At the first iteration, the poll set P_0 is represented by the four open circles. The trial point $[1, 2]^T$ has an objective function value that is less than $f(x_0)$, leading to a successful iteration. Iteration 1 starts with $x_1 = [1, 2]^T$. However, iteration 1 fails in identifying a trial point in the poll set P_1 with a lower objective function value. Thus, iteration 2 starts with $x_2 = x_1$, but the step size parameter is cut in half, so the new poll set P_2 uses closer points.

4.3.1 PS in DAKOTA

DAKOTA supports a Pattern Search algorithm with several options that allow the user to gain control over the process. The most useful and important ones to determine a successful optimisation are described in Table 4.3. These values will be used in the validation tests performed in Chapter 8 and Chapter 9.

Item	Characteristic
Initial point x_0	From global optimisation or user choice
Initial step size	$\delta_0 = 0.2$, 20% of the domain [a]
Contractor factor	$c = 0.5$ (DAKOTA default and recommended value)
Convergence	Step length-based: $\epsilon_{\text{stop}} = 1.E-5$ (DAKOTA default value)
	Based on objective function value (solution target, set by user)
Stopping criteria	Maximum number of function evaluation allowed: 1000 (DAKOTA default value)
	Maximum number of iterations allowed: 100 (DAKOTA default value)

Table 4.3: Summary of parameters used for PS in DAKOTA

[a] Remember: domain for our application is $[0,1]^d$ with d the number of variables

Design space exploration 5

In this chapter, both the description of the initial sampling algorithms is given (Section 5.1), as well as the infill sampling criteria (Section 5.2).

5.1 Sampling methods

In Section 2.1.1, when SBO workflow was presented along with its issues, the two main problems about sampling plans have risen: (i) How many points are sufficient and necessary to represent the domain fully? Moreover, what is the relation between this number and the number of variables? (ii) Which points to chose in the domain? What does this decision depend on? We tried to give an overview of the answers that other authors gave to these questions in Section 3.3. We saw that for the first question, no standard rule exists on what should be the sample size. Furthermore, for the second, the best possible methods seem to be the Halton sequence and Latin Hypercube.

5.1.1 Latin hypercube

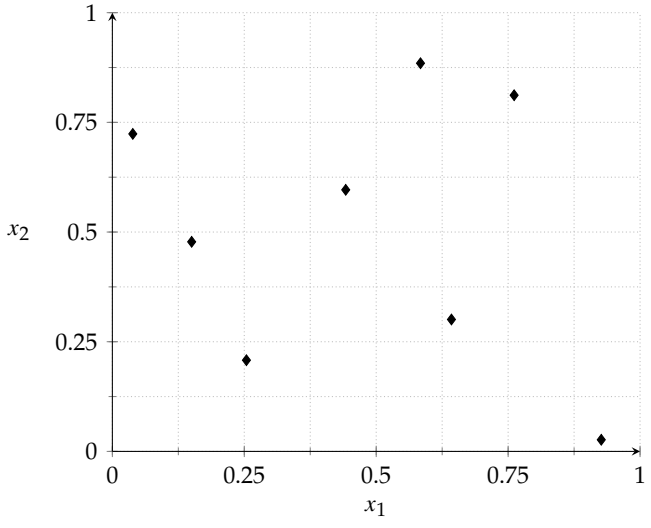


Figure 5.1: LHS good sampling example in $\mathbb{R}^2$ with $p = 8$ sampling points

Latin Hypercube Sampling (LHS) was first proposed by McKay et al. in 1979 [60] and it is a method that allows to explore the hyper-rectangle $[l, u] \subset \mathbb{R}^n$ partitioning the range of each variable into p intervals. For each variable there will be exactly one element of the sample set randomly selected in each of these p intervals. Let Π be a $n \times p$ matrix in which each of its n rows is a random permutation of

[60]: McKay *et al.* (1979), 'Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code'

the row vector $[1, 2, \cdots, p]$. Than set

$$x_i^j = l_i + \frac{(\Pi_{i,j} - r_{i,j})}{p} (u_i - l_i) \tag{5.1}$$

for each $i = 1, 2, \cdots, n$ and $j = 1, 2, \cdots, p$ and where $r_{i,j} \in \mathbb{R}$ is randomly chosen in the interval $[0, 1)]$ return $\{x^1, x^2, \cdots, x^p\} \subset \mathbb{R}^n$ [7].

Figures 5.1 and 5.2 shows three examples of LHS, using $p = 8$ and $p = 25$ points in $[0, 1] \times [0, 1] \subseteq \mathbb{R}^2$. In the figures it is visible how there is exactly one point in each line and in each column delimited by the lines.

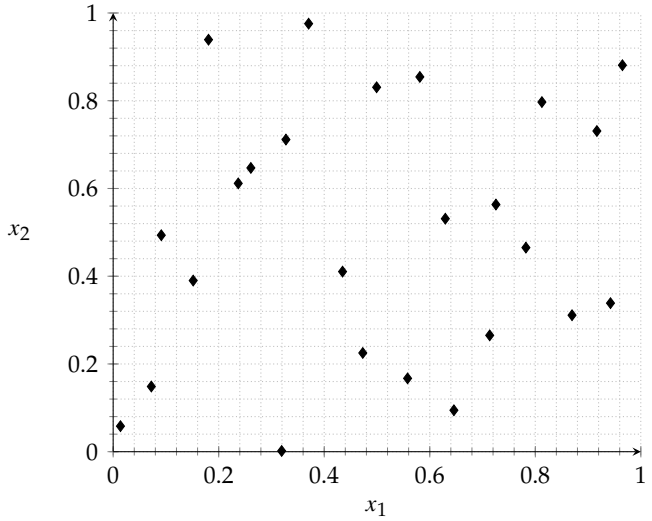


Figure 5.2: LHS sampling example in $\mathbb{R}^2$ with $p = 25$ sampling points

However, a drawback could be that the generated matrix is not repeatable because of the randomness. Also, the space-filling aspect of space produced by this technique is not guaranteed to be good all the time. Wang et al. (2007) [97] found that LHS is only uniform in 1-D projection. In fact we can see figure 5.3, compared to figure 5.1, how with the same premises and equations the space is not adequately covered.

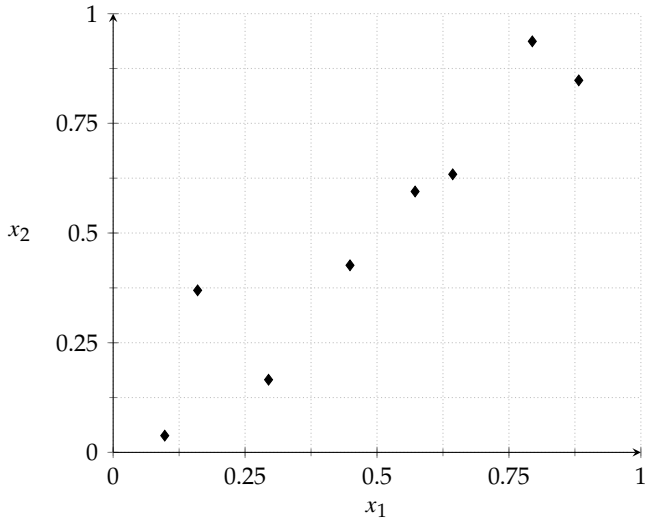


Figure 5.3: LHS bad sampling example in $\mathbb{R}^2$ with $p = 8$ sampling points

Therefore, a further improvement for a more uniform coverage

of the space involves maximising the Euclidean distance d_p between points i and j. For a n-dimension sampling the distance is defined in equation 5.2.

$$d_p(x_i, x_j) = \left(\sum_{b=1}^{a} |x_{jb} - x_{ib}|^2 \right)^{1/p} \tag{5.2}$$

for $p = 1$ this is the rectangular distance and $p = 2$ yields the *Euclidean norm*. This method provides generally good uniformity and flexibility on the size of the sample [44].

DAKOTA reference manual [3] states that Latin Hypercube Sampling is very robust and can be adjusted to any problem. It is reasonably effective at determining the mean of model responses and linear correlations with a moderately small number of samples relative to the number of variables. Moreover, it is possible to use random seed control. This provides a mechanism for making a stochastic method repeatable. That is, when the same seed is used in identical studies, the results will be identical.

5.1.2 Halton sequence

Halton sequence was firstly introduced by Halton in 1960 [31] . It is a quasi-random low-discrepancy sequence that can be used in k dimensions. The points in the sequence are specially designed to fill a $[0, 1)^k$ unit cube in a uniform way.

[31]: Halton (1960), 'On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals'

Let $R \geq 2$ be an integer, then any other integer $n \geq 0$ can be written in the radix-R notation as:

$$n = n_0 + n_1 R + n_2 R^2 + \cdots + n_M R^M \tag{5.3}$$

with $0 \leq R_i < n$ and where $M = [\ln n / \ln R]$ with square brackets denoting the integral part. By using a radical inverse function φ, we can uniquely construct a fraction lying between 0 and 1

$$\varphi_R(n) = n_0 R^{-1} + n_1 R^{-2} + \cdots + n_M R^{-M-1} \tag{5.4}$$

The Halton sequence in k-dimensions $\{\boldsymbol{x}_1\}$ is then defined as:

$$\boldsymbol{x}_n = (\varphi_{R_1}(n), \varphi_{R_2}(n), \cdots, \varphi_{R_k}(n)) \tag{5.5}$$

for $n = 1, 2, \cdots, N$.

In figure 5.4 we can see the first 25 points of the Halton sequence in the space $[0, 1)^2$. To generate this results, for x_1 the sequence was based on $R = 2$. This means that the horizontal space was firstly divided in half, then in fourths, eights, etc., generating the fractions (without repetitions):

$$\frac{1}{2}, \quad \frac{1}{4}, \quad \frac{3}{4}, \quad \frac{1}{8}, \quad \cdots \tag{5.6}$$

On the other hand, for x_2 the sequence was based on $R = 3$, meaning the vertical space was divided in thirds, ninths, twenty-sevenths, etc. This generated:

$$\frac{1}{3}, \quad \frac{2}{3}, \quad \frac{1}{9}, \quad \frac{4}{9}, \quad \cdots \tag{5.7}$$

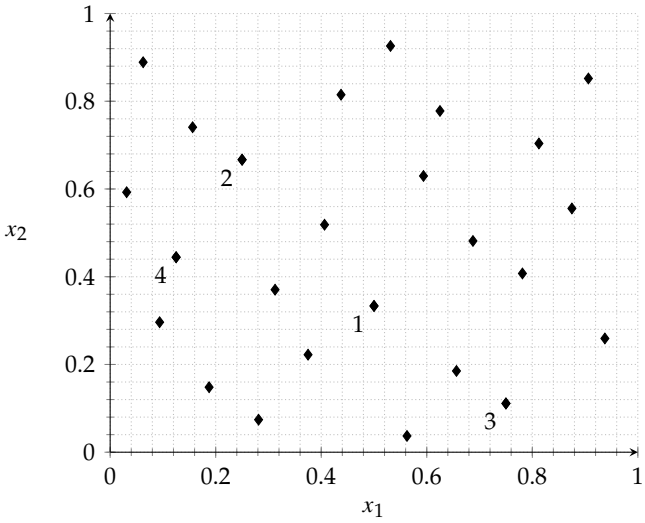


Figure 5.4: Halton sequence example in two dimensions with 25 sampling points

when pairing the fractions in 5.6 and 5.7, we obtain the coordinates of the points. Th first four points of the sequence are numbered in Figure 5.4 and are:

$$\left(\frac{1}{2}, \frac{1}{3}\right), \quad \left(\frac{1}{4}, \frac{2}{3}\right), \quad \left(\frac{3}{4}, \frac{1}{9}\right), \quad \left(\frac{1}{8}, \frac{4}{9}\right) \tag{5.8}$$

One of the advantages of the Halton sequence is its ease of implementation. Wang et al. (2007) [97] wrote that Hammersley, thus also Halton, sampling is found to provide a better uniformity than Latin Hypercube designs. However, for a low number of sampling points or a too high number of variables k, the method does not assure adequate space-filling.

5.2 Infill sampling criteria

In Section 2.1.1, when SBO workflow was presented along with its issues, the two main problems about infill sampling plans were risen: (i) How many points should be used in the refinement phase? Moreover, what is the ratio between this number and the number of initial sampling points? (ii) Which points to chose in the domain? How to balance a global and a local search? We tried to give an overview of the answers that other authors gave to these questions in Section 3.3. We saw that for the first question a common practice is to use approximately between a third and two-thirds of the total number of available points. Furthermore, for the second, the best possible methods seem to be the Expected Improvement. However, this can be only used with Kriging surrogate model. Thus, also, an alternative was selected.

5.2.1 Expected improvement

As already established in Section 3.3.3, the Expected Improvement (EI) algorithm is promising but can be applied only with a Gaussian process-based model, such as Kriging, discussed more in detail in

Section 6.3 or RBF, discussed in Section 6.2. Moreover, the reason is that this surrogate model gives as output, not just the prediction of value ($\widehat{f}$) for each point of the domain $\boldsymbol{x}$, but also an estimated error. The error is in the form of a standard deviation. We model the uncertainty of the prediction by considering it as the realisation of a normally distributed random variable $Y(\boldsymbol{x})$ with mean $\widehat{f} = \widehat{y}(\boldsymbol{x})$ (the most likely prediction) and a variance $\widehat{s}^2(\boldsymbol{x})$. In Section 6.3.1, the derivation of how to calculate it will be given.

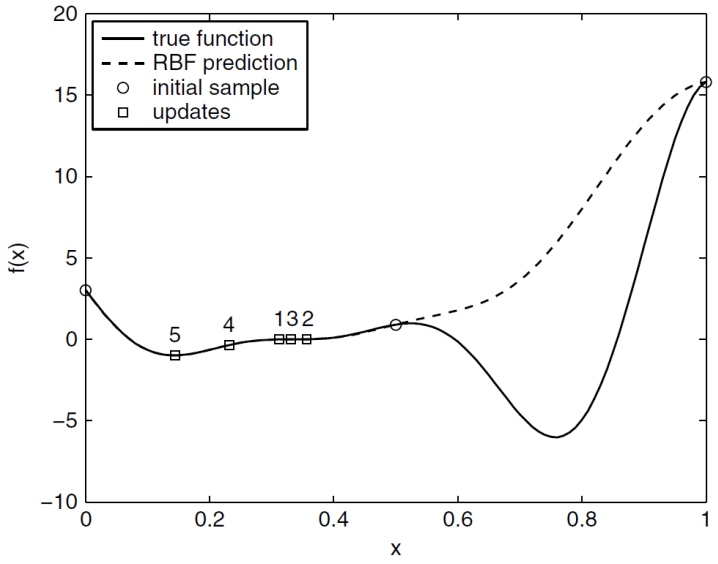


Figure 5.5: A RBF prediction of the true function $f(x) = (6x-2)^2 \sin(12x-4)$ with 3 initial samples and 5 refinement points. From Forrester et a. (2008) [23].

For now, it is only important to understand that when performing an infill strategy, the wish it to position the next point at the value of $\boldsymbol{x}$ that will lead to an improvement of the best observed data so far, $y_{\min}$. The Improvement is $I = y_{\min} - Y(\boldsymbol{x})$ and the Probability of Improvement (PoI) is defined as:

$$P[I(\boldsymbol{x})] = \frac{1}{\widehat{s}\sqrt{2\pi}} \int_{-\infty}^{0} \exp\left(-\frac{(I-\widehat{y}(\boldsymbol{x}))^2}{2\widehat{s}^2}\right) dI \tag{5.9}$$

But it is usually calculated using the error function:

$$P[I(\boldsymbol{x})] = \frac{1}{2}\left[1 + \operatorname{erf}\left(\frac{y_{\min} - \widehat{y}(\boldsymbol{x})}{\widehat{s}\sqrt{2}}\right)\right] \tag{5.10}$$

When $\widehat{s} = 0$ also $P[I(\boldsymbol{x})] = 0$, so there is no possibility of re-sampling. Equation 5.10 is interpreted graphically in Figures 5.5 and 5.6, taken from [23]. The first figure shows the Radial Basis Function prediction of the true function $f(x) = (6x-2)^2 \sin(12x-4)$ with 3 initial samples and 5 refinement samples selected with a minimum prediction based infill strategy (see Section 5.2.2).

The second represents the Probability of Improvement for the same prediction. The highest probability is in the local minimum found by RBF, but there is also a PoI in the region where points are most scarce. This Figure 5.6 indicates where an improvement might be found, but it does not show how big that improvement could be. To calculate the amount of improvement that we expect, given the mean $\widehat{y}(\boldsymbol{x})$ and

variance $\widehat{s}^2(\boldsymbol{x})$, we use the Expected Improvement (EI) function given by:

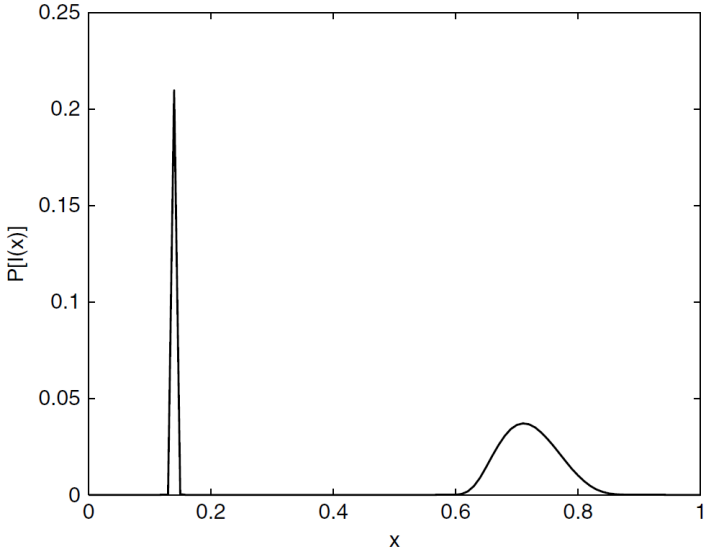


Figure 5.6: The probability of improvement in the prediction shown in Figure 5.5. From Forrester et al. (2008) [23].

$$E[I(\boldsymbol{x})] = \begin{cases} \left(y_{\min} - \widehat{y}(\boldsymbol{x})\right) \Phi\left(\frac{y_{\min} - \widehat{y}(\boldsymbol{x})}{\widehat{s}(\boldsymbol{x})}\right) + \widehat{s}\phi\left(\frac{y_{\min} - \widehat{y}(\boldsymbol{x})}{\widehat{s}(\boldsymbol{x})}\right) & \text{if } \widehat{s} > 0 \\ 0 & \text{if } \widehat{s} = 0 \end{cases} \tag{5.11}$$

where $\Phi(.)$ and $\phi(.)$ are the cumulative distribution function and the probability density function respectively. Like $P[I(\boldsymbol{x})]$, also $E[I(\boldsymbol{x})] = 0$ when $\widehat{s} = 0$. This is why a maximum expected improvement infill procedure will eventually find the global optimum [23]. Equation 5.11 can be evaluated using the error function:

$$E[I(\boldsymbol{x})] = \left(y_{\min} - \widehat{y}(\boldsymbol{x})\right)\left[\frac{1}{2} + \frac{1}{2}\text{erf}\left(\frac{y_{\min} - \widehat{y}(\boldsymbol{x})}{\widehat{s}\sqrt{2}}\right)\right] +$$

$$+ \widehat{s}\frac{1}{\sqrt{2\pi}} \exp\left[\frac{-(y_{\min} - \widehat{y}(\boldsymbol{x}))^2}{2\widehat{s}^2}\right)\Bigg] \tag{5.12}$$

Plotting the EI function for the prediction in Figure 5.5 for all values of $\boldsymbol{x}$ yields the plot in Figure 5.7. In contrast to the PoI in Figure 5.6, the EI is greatest in the unsampled area of the global minimum. This is because, although there is a high probability of some improvement at the point that maximises $P[I(\boldsymbol{x})]$, the actual amount of improvement is likely to be greater at the point that maximises $E[I(\boldsymbol{x})]$.

In conclusion, for most applications maximising $E[I(\boldsymbol{x})]$ will provide the best route to finding the global optimum. However, the method could be tricked by a particularly weak or unlucky initial sample and a very deceptively positioned optimum [23].

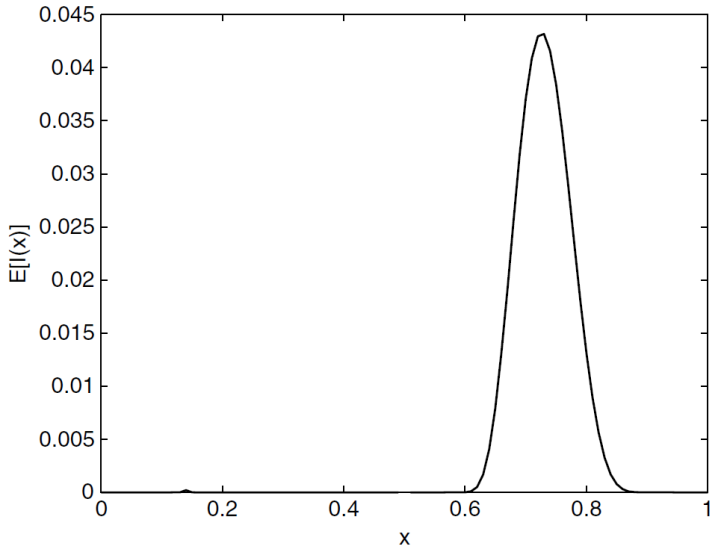


Figure 5.7: The Expected Improvement in the prediction shown in Figure 5.5. From Forrester et al. (2008) [23].

EI in DAKOTA

DAKOTA implementation of the Expected Improvement function makes use of the so-called Efficient Global Optimisation (EGO), firstly developed by Jones, Schonlau, and Welch [47]. There are two major differences between DAKOTA implementation and that of [47]: DAKOTA uses a DIRECT [8] algorithm to find points which maximise the EI. Second, multi-objective optimisation and nonlinear constraints are allowed. For more information see DAKOTA's Theory [4] and User [5] Manuals. The general procedure for this EGO-type method is:

8: A DIvision of RECTangles (DIRECT) algorithm adaptively subdivides the space of feasible design points to guarantee that iterates are generated in the neighbourhood of a global minimum in a finite number of iterations.

1. Build an initial Gaussian process model of the objective function;
2. Find the point that maximises the EI function. If the relative change of the new point compared to the previous is sufficiently small, stop;
3. If the EI function value at this point is sufficiently small, stop;
4. Evaluate the true objective function at the point where the EI function is maximised. Update the Gaussian process model using this new point;
5. Return to the previous step.

As an example, Figure 5.8 shows the process of a search of the test function $f(x) = (6x-2)^2 \sin(12x-4)$ in the range $[0,1]$ using a maximum $E[I(\boldsymbol{x})]$ infill strategy starting from an initial sample of three points. Kriging meta-model and DAKOTA toolkit were used. We can see that to begin; the search isolates the local optimum to the left of the plot. However, afterwards, there is still an expectation of improvement to the right, and so the global optimum is found in nine steps.

In conclusion, there are three possible stopping criteria for this method: a $\boldsymbol{x}$-convergence tolerance, an Expected Improvement convergence tolerance, and a maximum number of iterations allowed.

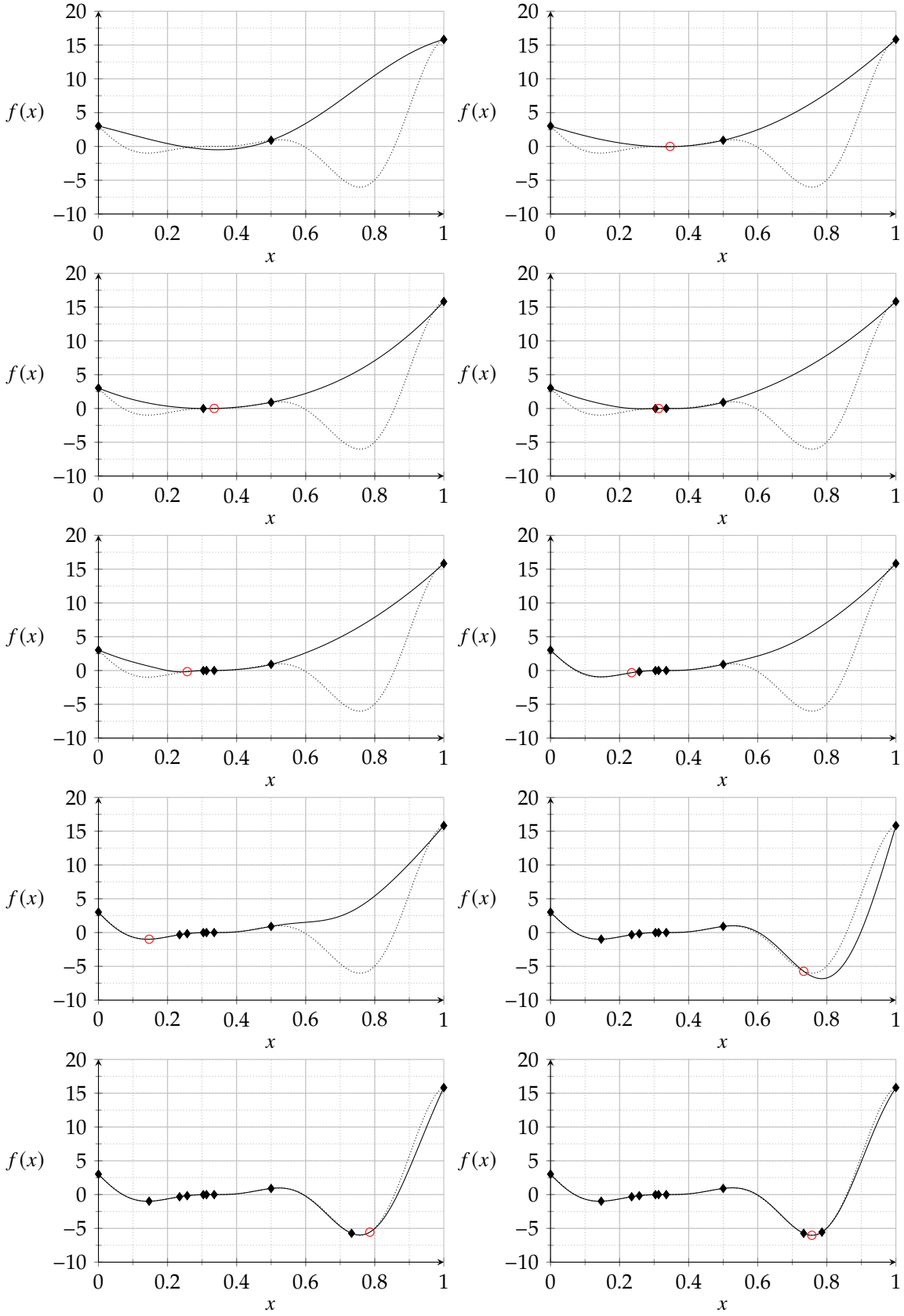


Figure 5.8: The progress of a search of the one-variable test function $f(x) = (6x-2)^2 \sin(12x-4)$ in the range $[0,1]$ using a maximum $E[I(\boldsymbol{x})]$ infill strategy

5.2.2 Minimisation of $\widehat{f}$

This method is one of the most common when performing local exploitation. It works in an iterative scheme where optimisation is performed on a surrogate level. Given a set of points where the true function $f(\boldsymbol{x})$ is known, each iteration:

1. The optimal solution of the surrogate model is found. Then this solution is passed to the next iteration;
2. The optimum surrogate point is evaluated with the truth model. This solution is added back to the set of points upon which the next surrogate is constructed.

This way, the optimisation acts on a more accurate surrogate model during each iteration, presumably driving to optimality. However, this method has a high risk of missing the global minimum, especially when the initial sampling plan was somehow deceiving or scarce.

Minimisation of $\widehat{f}$ in DAKOTA

This method in DAKOTA is called Surrogate Based Global (SBG). In each iteration, the optimum is searched in the whole domain, as opposed to Surrogate Based Local (SBL) where each iteration also a different trust region is defined. It was originally designed for Multi-Objective Optimisation via Genetic Algorithm. Some cautionary notes are given in the User's manual [5] when approaching this method:

- This approach has no guarantee of convergence;
- One might first try a single minimisation method coupled with a surrogate model before using the surrogate-based global method. This is essentially equivalent to setting the maximum number of iterations to 1 and will allow one to get a sense of what surrogate types are the most accurate to use for the problem;
- Also note that one can specify that surrogates be built for all primary functions and constraints or only a subset of these functions and constraints. This allows one to use a "truth" model directly for some of the response functions, perhaps due to them being much less expensive than other functions;
- We initially recommend a small number of maximum iterations, such as 3-5, to get a sense of how the optimisation is evolving as the surrogate gets updated. If it appears to be changing significantly, then a larger number (used in combination with restart) may be needed.

In conclusion, this method alone is not enough to capture a global optimum inside the domain. Its performance could be enhanced by adding points in areas of scarce sampling (exploration). Furthermore, from the recommendations in the DAKOTA manual, it looks like this implementation of the method is not suitable for an automated routine.

5.2.3 Distance approach

When performing a global exploration of the design space, one of the most used method is the Maximin Distance Approach. This method usually consists of four steps. Given a set of points where the true function $f(\boldsymbol{x})$ is known, each iteration:

1. A surrogate model is constructed;
2. A candidate set is created and scored based on the maximin function;
3. The best point from the set is selected to be evaluated with the truth model. Given the existing sample set $\boldsymbol{x}_P$ of l points, the Maximin Distance approach is to select a new sample set $\boldsymbol{x}_C$ of m points to maximise the minimum distance between any two sample points in the sample set $\boldsymbol{x}_A = \boldsymbol{x}_C \cup \boldsymbol{x}_P$ of $(l+m)$ points such that:

$$\max_{\boldsymbol{x}_C}\left[\min_{\substack{\boldsymbol{x}_{Ci}\leq\boldsymbol{x}_{Aj}\\ 1\leq i\leq m,\,1\leq j\leq(l+m)}}\left(d(\boldsymbol{x}_{Ci},\boldsymbol{x}_{Aj})\right)\right] \tag{5.13}$$

   with d the Euclidean distance, defined in 5.2;
4. This new point is added to the set of true points and the approach returns to step 1, unless a maximum number of evaluations is reached.

Distance approach in DAKOTA

DAKOTA has a method available called Adaptive Sampling (AS). It can generate one or a batch of points to add at the time based on scores given to each candidate. A candidate's score is the Euclidean distance in domain space between the candidate and its nearest neighbour in the set of points already evaluated on the exact model. Therefore, the most undersampled area of the domain will always be selected. Note that this is a space-filling metric. However, the adaptivity of this method could be brought to question as it would choose the same points regardless of the surrogate model used.

Unfortunately, by the time this thesis is written, this is an experimental capability, even though a new DAKOTA version is about to be released. Thus, another possibility to perform a global exploration of the space is to add more points to the initial sampling plan with a space-filling technique, like the ones described in Section 5.1.

Surrogate models 6

In Section 2.1.1, when SBO workflow was presented along with its issues, the three main problems about surrogate models have risen: (i) Which factors influence the choice of a meta-model over the others? What are the factors that characterise a common case at Van Oossanen N.A.? (ii) How to assess the model accuracy? Furthermore, which factors play a role in measuring their performance? (iii) How to properly visualise the surrogate models? We tried to give an overview of the answers that other authors gave to these questions in Section 3.4. The results are summarised again in Table 6.1.

Table 6.1: Summary of conclusions drawn from the surrogate model literature review

Issue	Conclusion
Factors that influence the choice	Quality of data available Dimensionality of solution space Complexity of solution surface Degree of noice associated with data Ease of use
Model accuracy	Performance measured through: – Accuracy – Robustness and flexibility – Efficiency – Transparency – Conceptual simplicity
	Metrics – RMSE – MaxAE – CV and LOO – CPU time
Visualisation tools	Histogram and scatterplots of variables x and objectives f Plots of approximated function $\widehat{f}$ over each variable

The three most promising meta-models that we are going to analyse more in detail are Artificial Neural Network, Radial Basis Functions, and Kriging.

6.1 Artificial neural network

Artificial Neural Networks (ANN) can be described as a big parallel, interconnected network of basic computing elements that process information imitating the functioning of the brain. Analogous to the

biological model, which consists of a large number of neurons, ANN has many similarly connected computational elements referred to as artificial neurons or nodes. A neural network is created by assembling neurons into a network architecture [30].

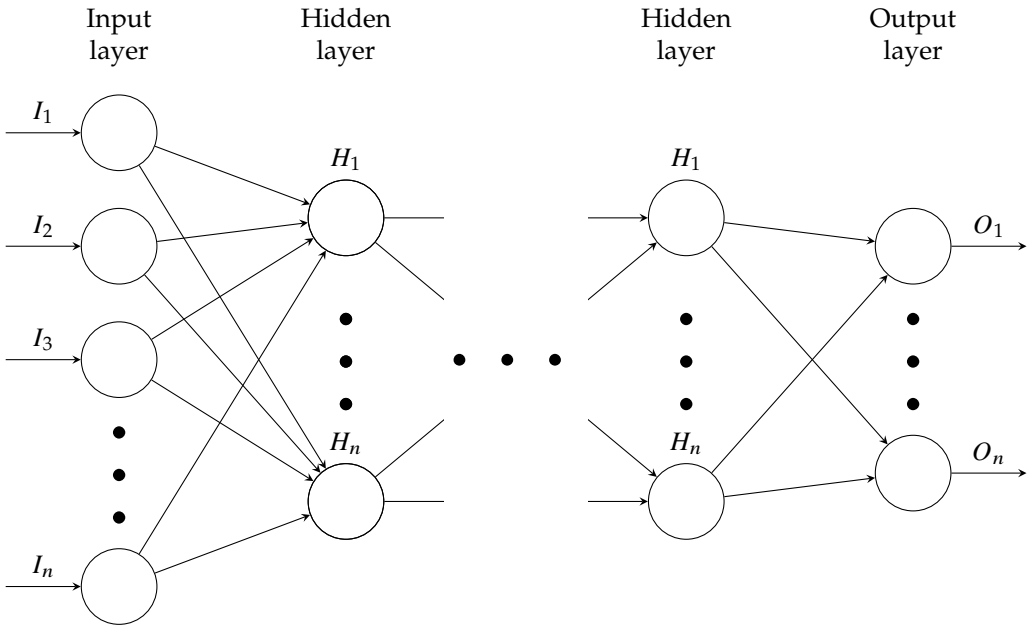


Figure 6.1: ANN diagram

The form with widest application in the area of meta-modelling is the Feed-Forward Neural Network (FFNN), in particular Multi-Layer Perceptron (MLP) [57] [74] [86]. It consists of a series of input nodes, connected to output nodes via one or more intermediate layers, called hidden layers (Figure 6.1). Also, each neuron in a layer is connected to every neuron on the next one but neither connections between units in the same level nor closed-loops, feedback, are allowed. The neural network is trained by presenting it pairs of input and output data and then iteratively adjusting weights in the connections between computing elements so that its output matches the known output data. Once trained, the networks can be used to replace complex and time-consuming analysis procedures [8].

The number of hidden layers and elements in the network can vary, and finding the optimal network architecture for fitting a given data-set is a non-trivial problem. Having too many hidden neurons means the system is overspecified, and incapable of generalisation, on the other hand having too few can prevent the system from adequately fitting the input data, and reduces its robustness [56].

The input data is used to compute the output of each unit through an activate function. In general, it is a non-linear fixed function that must be chosen by the user. Figure 6.2 displays its functioning and its generic formulation is shown in equation 6.1.

$$\hat{y} = \Psi\left(\sum_{i=1}^{k} w_i x_i + b\right) \tag{6.1}$$

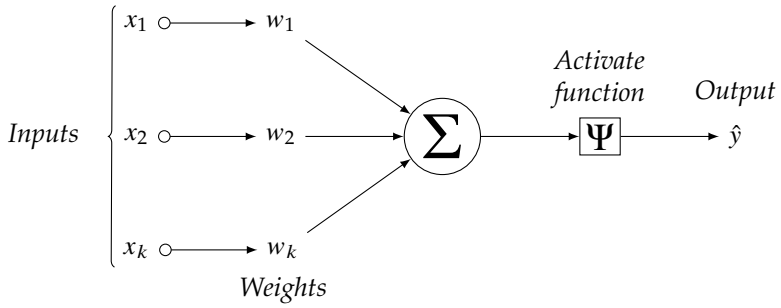


Figure 6.2: Activation function operation flow chart

where k is the number of sample points, x_i are the inputs of the neurons, w_i are the corresponding weights of the i^{th} input. Ψ is the activation function and b is a possible source of noise. Back-propagation is one of the most commonly used learning algorithms to determine the weights which minimises the global error by gradient decent [77] .

[77]: Rumelhart *et al.* (1986), 'Learning representations by back-propagating errors'

ANNs ensure few advantages when applied to pattern recognition problems. No prior assumptions about the data distribution are needed as ANNs adjust themselves to the particular problem constraints during the learning process. Also, according to Sarle (1994) [80], they are general-purpose, flexible, non-linear models that, given enough hidden neurons and enough data, can approximate virtually any function to any desired degree of accuracy.

However, as ANN learn by example, it requires a large amount of data to be trained appropriately, and the user requires some knowledge of how to select and prepare the data [56]. Neural networks are highly prone to the risk of over-fitting [74].

Pros	Cons
– No prior assumptions about data distribution are needed	– It requires a large amount of data to be trained
– Very flexible for different problems	– The user must have some knowledge of how to select and prepare the data
– Can deal with non-linear data and large problems	

Table 6.2: Pros and cons of Artificial Neural Network

6.1.1 ANN in DAKOTA

The ANN surface fitting method in Dakota employs a Stochastic Layered Perceptron (SLP) artificial neural network based on the direct training approach of Zimmerman [104]. The SLP ANN method is designed to have a lower training cost than traditional back-propagation neural networks. The form of the SLP ANN model is shown in equation 6.2. The hyperbolic tangent function is a common activation function since it has been demonstrated that it provides fast convergence of training algorithms.

$$\hat{f}(\boldsymbol{x}) \approx \tanh(\tanh((\boldsymbol{x}\boldsymbol{A}_0 + \theta_0)\boldsymbol{A}_1 + \theta_1)) \tag{6.2}$$

where x is the current point in n-dimensional parameter space, and the terms $\boldsymbol{A}_0$, θ_0, $\boldsymbol{A}_1$, θ_1 are the matrices and vectors that correspond to the neuron weights and offset values in the ANN model. These terms are computed during the ANN training process and are analogous to the polynomial coefficients in a quadratic surface fit. A singular value decomposition method is used in the numerical methods that are employed to solve for the weights and offsets.

The SLP ANN is a non-parametric surface fitting method. Thus, it can be used to model data trends that have slope discontinuities as well as multiple maxima and minima. However, the ANN surface is not guaranteed to exactly match the response values of the data points from which it was constructed [5].

Three non-compulsory options can be used in Dakota to modify the characteristics of SLP ANN [3]. They are:

Correction factors, which force the surrogate models to match the true function values and possibly true function derivatives at the centre point of each trust region. Currently, Dakota supports either zeroth-, first-, or second-order accurate correction methods, each of which can be applied using either an additive, multiplicative or combined correction function. The default behaviour is that no correction factor is applied.

Max_nodes that limits the maximum number of hidden layer nodes in the neural network model. The default is to use one less node than the number of available training data points yielding a fully-determined linear least squares problem. However, reducing the number of nodes can help reduce over-fitting and more importantly, can drastically reduce surrogate construction time when building from a large data set.

Range that controls the range of the input layer random weights in the neural network model. The default range is 2.0, resulting in weights in $(-1, 1)$.

6.2 Radial basis functions

Radial Basis Functions (RBF) have been developed for scattered multivariate data interpolation (Hardy, 1971 [33]; Dyn, et al., 1986 [18]). It is a model that from the sampling plan $\boldsymbol{x} = \{x^{(1)}, x^{(2)}, \cdots, x^{(n)}\}^T$, yielding the responses $\boldsymbol{y} = \{y^{(1)}, y^{(2)}, \cdots, y^{(n)}\}^T$, uses a linear combination of weighted basis functions whose value only depends on the Euclidean distance $r^{(i)} = \| \boldsymbol{x} - \boldsymbol{c}^{(i)} \|$. Here $\boldsymbol{c}^{(i)}$ is the sampling point position and $\boldsymbol{x}$ is the prediction site. The formulation of this model is identical to that of a single-layer neural network with radial coordinate neurons [23]. The approximation to the design space $\widehat{f}$ is given in equation 6.3

$$\widehat{f}(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{\psi} = \sum_{i=1}^{m} w_i \psi \left(\| \boldsymbol{x} - \boldsymbol{c}^{(i)} \| \right) \tag{6.3}$$

where $\boldsymbol{w}$ are the weights, one for each function, and $\boldsymbol{\psi}$ is the m-vector containing the values of the basis functions $\psi(r)$. Examples of commonly used basis functions, with the main feature of having a response that decreases (or increases) monotonically with distance from the central point, are:

- Cubic:

$$\psi(r) = r^3 \tag{6.4}$$

- Gaussian:

$$\psi(r) = e^{-r^2/(2\sigma^2)} \tag{6.5}$$

- Multi-quadratic or inverse multi-quadratic:

$$\psi(r) = (r^2 + \sigma^2)^{\pm 1/2} \tag{6.6}$$

where σ^2 is the variance: a constant shape parameter greater than zero that represents the sensitivity, or spread, of the basis function over the input domain. Usually, this value is assumed identical for all functions. Figure 6.3 shows three examples of one-dimensional Gaussian functions, with its characteristic symmetric "bell curve" shape. Each curve has the same peak centre position ($c = 0$), but different variance σ^2.

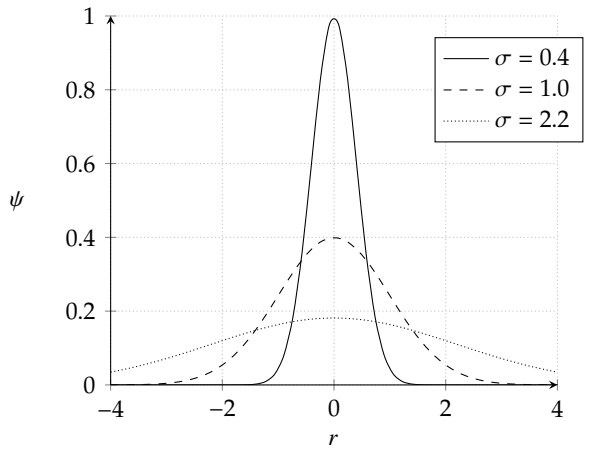


Figure 6.3: Examples of Gaussian Distribution functions

Whether a set of parametric basis functions or fixed ones are chosen, the weights $\boldsymbol{w}$ are easy to estimate. The fundamental condition is that in the n points where the responses $\boldsymbol{y}$ are known, the predictor $\widehat{f}$ gives the same values. This can be done via the interpolation condition in equation 6.7

$$\widehat{f}(\boldsymbol{x}^{(j)}) = \sum_{i=1}^{m} w_i \psi\Big(\|\boldsymbol{x} - \boldsymbol{c}^{(i)}\|\Big) = y^{(j)}, \qquad j = 1, \cdots, n \tag{6.7}$$

Equation 6.7 is linear in terms of the basis function weights $\boldsymbol{w}$, yet the predictor $\widehat{f}$ can express highly non-linear responses [22]. One of the conditions of obtaining a unique solution is that the system 6.7 must be "square", that is $m = n$. It also simplifies things if the bases actually coincide with the data points ($\boldsymbol{c}^{(i)} = \boldsymbol{x}^{(i)}, \quad \forall\, i = 1, \cdots, n$). This leads to the matrix equation

$$\boldsymbol{\Psi w} = \boldsymbol{y} \tag{6.8}$$

where $\boldsymbol{\Psi}$ denotes the so-called *Gram Matrix*, defined as

$$\boldsymbol{\Psi}_{i,j} = \psi\left(\|\boldsymbol{x}^{(i)} - \boldsymbol{x}^{(j)}\|\right), \qquad i, j = 1, \cdots, n \tag{6.9}$$

The solution for the method is therefore the definition of weights via the computation of $\boldsymbol{w} = \boldsymbol{\Psi}^{-1}\boldsymbol{y}$ and the choice of which basis function to use is important to insure a safe computation. For example, it can be shown that, under certain assumptions, Gaussian and inverse multi-quadratic basis functions always lead to a symmetric positive definite Gram Matrix [93], ensuring safe computations.

Beyond determining $\boldsymbol{w}$, another task is to estimate other parameters introduced by the basis functions, such as σ. While the correct choice of $\boldsymbol{w}$ will make sure that the approximation can reproduce the training data, the correct estimation of these additional parameters will enable us to minimise the (estimated) generalisation error of the model [23].

It is also worth mentioning that very close proximity of any two points in $\boldsymbol{x}$ can cause ill-conditioning [62]. This is a rather unlikely event if $\boldsymbol{x}$ is a space-filling sampling plan, but can become a nuisance if clusters of infill points (see chapter Chapter 5) are added subsequently in specific areas of interest within the design domain.

In conclusion, Radial Basis Function models have satisfactory flexibility, simple structures, relatively few calculations and high-efficiency [43]. Another advantage of the model is its ability to fit many different functions, because of the freedom to choose different basis functions and different values for the weights [71]. Moreover, it has a closed-form solution, which allows the model to be robust and cheap to compute, and its application is relatively straightforward, as no parameters need to be specified by a user [44].

However, a weakness of the model is that if the responses $\boldsymbol{y}$ are corrupted by noise, the above equations may yield a model that over-fits the data, that is it does not discriminate between the underlying response and the noise [22]. Moreover, over-fitting [10] may also occur. The risk of over-fitting is higher when there are very few design sites relative to the number of parameters to be tuned [74]. Finally, RBF ability to model highly non-linear surfaces is limited when the number of sampling points is not high enough.

10: Over-fitting occurs when the model is, in some sense, too flexible and it fits the data at a too fine-scale, that is it fits the noise, as well as the actual underlying behaviour we are seeking to model [23]

Table 6.3: Pros and cons of Radial Basis Function

Pros	Cons
– Ok with non-linear problems (but with enough training points)	– It requires a high number of sampling points (for highly non-linear problems)
– Very robust	– Sensitive to noise
– Closed form solution	
– Cheap to compute	

6.2.1 RBF in DAKOTA

The only basis function supported in DAKOTA is the Gaussian one already described in equation 6.5, therefore we can write for the design space approximation:

$$\widehat{f}(\boldsymbol{x}) = \sum_{i=1}^{n} w_i \cdot \exp\left(\frac{-(r^{(i)})^2}{2\sigma^2}\right) \tag{6.10}$$

DAKOTA implementation determines the weights via a linear least squares solution approach. See [66] for more details. Four optional options can be used in Dakota to modify the characteristics of RBF [3]. They are:

[66]: Orr (1996), 'Introduction to radial basis function networks'

Correction factors, see Section 6.1.1.

Bases that modify the initial number of radial basis functions m. The default value is the smaller of the number of training points and 100. Thus caution is recommended when the number of training points n is higher than 100, and this value is not accordingly changed, as the condition of obtaining a unique solution $m = n$ would not be met.

Max_pts that control the maximum number of points to use in generating each RBF centre. Default value is $10 \cdot$ [bases]. Reducing this number will reduce the model building time.

Max_subset, which is the number of passes to take to identify the best subset of basis functions to use. Defaults to the smaller of $3 \cdot$ [bases] and 100.

6.3 Kriging

Kriging (KG) is named after the work in 1951 of Krige, a South African mining engineer, addressing problems in geostatistics [51]. More recently, Sacks et al. [79] [78], and Jones et al. [47] made it popular in the context of the modelling, and optimisation of deterministic functions, respectively. Starting from a set of sample data, $\boldsymbol{X} = \{\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \cdots, \boldsymbol{x}^{(n)}\}^T$, with observed responses, $\boldsymbol{y} = \{y^{(1)}, y^{(2)}, \cdots, y^{(n)}\}^T$, the goal is to find an expression for a predicted value at a new point $\boldsymbol{x}$. The Kriging prediction is based on the combination of a polynomial model and departures of the form:

$$\widehat{y}(\boldsymbol{x}) = \widehat{\mu}(\boldsymbol{x}) + Z(\boldsymbol{x}) + \varepsilon(\boldsymbol{x}) \tag{6.11}$$

where $\widehat{y}$ is the approximated unknown function of interest, and $\widehat{\mu}$ (the circumflex denotes a maximum likelihood estimate, MLE) is a mean base term that provides a "global" model of the design space. The general case, where $\widehat{\mu}$ is not restricted, is called *universal Kriging*. In many cases, however, it is simply taken to be a constant term [86] and this is often called *ordinary Kriging*. ε represents the approximation

error and Z is a set of n random vectors:

$$Z(\boldsymbol{x}) = \left(Z(\boldsymbol{x}^{(1)}) \cdots Z(\boldsymbol{x}^{(n)})\right)^T \tag{6.12}$$

11: $\mathbf{1}$ is a $n \times 1$ column vector of ones

The random field is built with $\mathbf{1}\mu$ [11] mean. The functions have variance σ^2, and non-zero covariance. A $n \times n$ correlation matrix $\boldsymbol{\Psi}$ is constructed as:

$$\boldsymbol{\Psi} = \begin{pmatrix} \text{corr}[Z(\boldsymbol{x}^{(1)}), Z(\boldsymbol{x}^{(1)})] & \cdots & \text{corr}[Z(\boldsymbol{x}^{(1)}), Z(\boldsymbol{x}^{(n)})] \\ \vdots & \ddots & \vdots \\ \text{corr}[Z(\boldsymbol{x}^{(n)}), Z(\boldsymbol{x}^{(1)})] & \cdots & \text{corr}[Z(\boldsymbol{x}^{(n)}), Z(\boldsymbol{x}^{(n)})] \end{pmatrix} \tag{6.13}$$

A variety of correlation functions can be chosen [85], however, the Gaussian correlation basis function proposed in Sacks, et al. in 1989 [79] is the most frequently used. Here k is the number of dimensions in the search space.

$$\text{corr}[Z(\boldsymbol{x}^{(i)}), Z(\boldsymbol{x}^{(l)})] = \psi^{(i)} = \exp\left(-\sum_{j=1}^{k} \theta_j \|x_j^{(i)} - x_j^{(l)}\|^{p_j}\right) \tag{6.14}$$

In particular, correlation and covariance are related as reported in equation 6.15:

$$\text{cov}[Z, Z] = \sigma^2 \text{corr}[Z, Z] \tag{6.15}$$

With the covariance matrix $\boldsymbol{C}$ defined as:

$$\boldsymbol{C} = \text{cov}[Z, Z] = \begin{pmatrix} \text{cov}[Z^{(1)}, Z^{(1)}] & \cdots & \text{cov}[Z^{(1)}, Z^{(n)}] \\ \vdots & \ddots & \vdots \\ \text{cov}[Z^{(n)}, Z^{(1)}] & \cdots & \text{cov}[Z^{(n)}, Z^{(n)}] \end{pmatrix} \tag{6.16}$$

Covariance is a measure of the correlation between two or more sets of random variables. This assumed correlation between sample data reflects the expectation that an engineering function will be smooth and continuous [23]. It is defined in general as:

$$\text{cov}[X, Y] = E[(X - \mu_X)(Y - \mu_Y)] = E[XY] - \mu_X \mu_Y \tag{6.17}$$

where μ_X and μ_Y are the means of X and Y and E is the expectation.

Looking at equation 6.14, we can see similarities with Gaussian basis function introduced for RBF model in equation 6.5. Where a Gaussian radial basis function has $1/\sigma^2$, the Kriging basis has a vector $\boldsymbol{\theta} = \{\theta_1, \theta_2, \cdots, \theta_k\}^T$, allowing the width of of the basis function to vary from variable to variable along the k-th direction. An example of how it works is given is figure 6.4. It shows how the choice of θ_j affects the correlation. It is essentially a width parameter that affects how far a simple point's influence extends. A low θ_j means a high correlation, with $Z(x_j^{(i)})$ being similar across the sample. Therefore θ_j can be considered as a measure of how active the function we are approximating is [23].

Also, where in the Gaussian basis (Eq. 6.5) the exponent is fixed at 2, giving a smooth function through the point $\boldsymbol{x}^{(i)}$, Kriging (Eq.

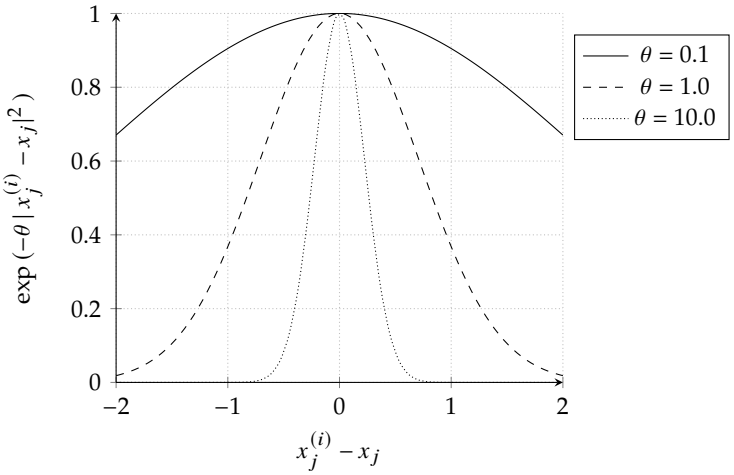


Figure 6.4: Examples of Correlations with varying θ

6.14) allows this exponent $\boldsymbol{p}_j = \{p_1, p_2, \cdots, p_k\}^T$ to vary. Typically $p_j \in [1,2]$ for each dimension in $\boldsymbol{x}$. With $\boldsymbol{p}$ fixed at $p_{(1,2,\cdots,k)} = 2$ and with constant θ_j for all dimensions, the Kriging basis function is in fact the same as the Gaussian. An example of how $\boldsymbol{p}$ works is given is figure 6.5. It shows how it affects the smoothness of the correlation function. With $p_j = 2.0$ there is a smooth correlation with a continuous gradient through $x_j^{(i)} - x_j = 0$. Reducing p_j increases the rate at which the correlation initially drops as $|x_j^{(i)} - x_j|$ increases. With a very low value of $p_j = 0.1$ there is no immediate correlation between the two points and there is a near discontinuity between $Z(\boldsymbol{x}_j^{(i)})$ and $Z(\boldsymbol{x}_j)$ [23].

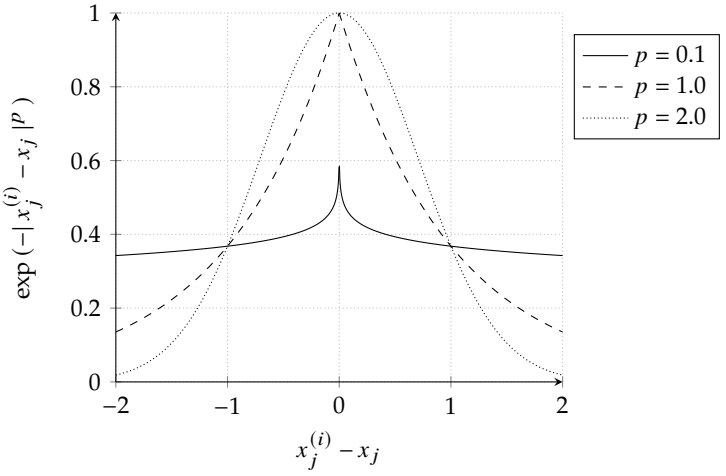


Figure 6.5: Examples of Correlations with varying p

The hyperparameters $\boldsymbol{p}$ and $\boldsymbol{\theta}$ are estimated via the Maximum Likelihood Estimation (MLE) of $\boldsymbol{y}$. This has the purpose of minimising the generalisation error of the model. The likelihood function for this Gaussian process expressed in terms of the sample data is:

$$L = \frac{1}{(2\pi\sigma^2)^{n/2}\,|\Psi\,|^{1/2}} \exp\left(-\frac{(\boldsymbol{y} - \mathbf{1}\mu)^T\ \boldsymbol{\Psi}^{-1}\,(\boldsymbol{y} - \mathbf{1}\mu)}{2\sigma^2}\right) \tag{6.18}$$

To simplify the likelihood maximisation, the natural logarithm can be

used. Therefore equation 6.19 has to be maximised:

$$\ln(L) = -\frac{n}{2}\ln(2\pi) - \frac{n}{2}\ln(\sigma^2) - \frac{1}{2}\ln|\Psi| - \frac{(\boldsymbol{y}-\mathbf{1}\mu)^T\,\boldsymbol{\Psi}^{-1}\,(\boldsymbol{y}-\mathbf{1}\mu)}{2\sigma^2} \tag{6.19}$$

By taking the derivatives of equation 6.19 and setting to zero, we obtain maximum likelihood estimates (i.e. best guesses) for μ and σ^2:

$$\widehat{\mu} = \frac{\mathbf{1}^T\,\boldsymbol{\Psi}^{-1}\,\boldsymbol{y}}{\mathbf{1}^T\,\boldsymbol{\Psi}^{-1}\,\mathbf{1}} \tag{6.20}$$

$$\widehat{\sigma}^2 = \frac{(\boldsymbol{y}-\mathbf{1}\mu)^T\,\boldsymbol{\Psi}^{-1}\,(\boldsymbol{y}-\mathbf{1}\mu)}{n} \tag{6.21}$$

These MLEs can now be substituted back into equation 6.19 and constant terms removed to give what is known as the concentrated ln-likelihood function. The value of this function depends only on unknown parameters $\boldsymbol{\theta}$ and $\boldsymbol{p}$.

$$\ln(L) \approx -\frac{n}{2}\ln(\widehat{\sigma}^2) - \frac{1}{2}\ln|\Psi| \tag{6.22}$$

Maximising equation 6.22, we get an estimate of the hyperparameters and Equations 6.20 and 6.21 are used to estimate $\widehat{\mu}$ and $\widehat{\sigma}^2$. Eventually, the predicted approximated function $\widehat{y}$ is given by:

$$\widehat{y} = \widehat{\mu} + \boldsymbol{\psi}^T\,\boldsymbol{\Psi}^{-1}\,(\boldsymbol{y}-\mathbf{1}\widehat{\mu}) \tag{6.23}$$

where the basis functions are contained in the vector $\boldsymbol{\psi}$. Also, the model is constructed in such a way that the prediction goes through all the data points. In this case $\boldsymbol{\psi}$ is the i-th column of $\boldsymbol{\Psi}$ and thus $\boldsymbol{\psi}\boldsymbol{\Psi}^{-1}$ is the i-th unit vector. Thus:

$$\widehat{y}(\boldsymbol{x}) = \widehat{\mu} + y^{(i)} - \widehat{\mu} = y^{(i)} \tag{6.24}$$

6.3.1 KG uncertainty

The Mean Square Error (MSE) in a Gaussian process based prediction such as Kriging (KG) is defined as:

$$\widehat{s}^2(\boldsymbol{x}) = \sigma^2\left(1 - \boldsymbol{\psi}^T\boldsymbol{\Psi}^{-1}\boldsymbol{\psi} + \frac{1-\mathbf{1}^T\boldsymbol{\Psi}^{-1}\boldsymbol{\psi}}{\mathbf{1}^t\boldsymbol{\Psi}^{-1}\mathbf{1}}\right) \tag{6.25}$$

The derivation of this equation can be found in Sacks et al. (1989) [79]. Equation 6.25 provides an estimate of the accuracy of the

The third term inside the parenthesis, which is due to uncertainty in the estimate of μ, is very small and is often omitted [23]. Always in equation 6.25, the first component in the parenthesis is the generalised least square prediction at a specific point given the design covariance matrix $\boldsymbol{C}$, while the second component pulls the generalised least square response surface through the observed points. We have seen how the elasticity of the response surface is determined by the correlation function $\psi^{(i)}$ [49].

In fact, if we are calculating $\widehat{s}^2(\boldsymbol{x})$ at a sample point $\boldsymbol{x}^{(i)}$, $\boldsymbol{\psi}\boldsymbol{\Psi}^{-1}$ is the i-th unit vector. Thus

$$\boldsymbol{\psi}^T\boldsymbol{\Psi}^{-1}\boldsymbol{\psi} = \boldsymbol{\psi}^T\mathbf{1} = \psi^{(i)} = \exp\left(-\sum_{j=1}^{k}\theta_j\|x_j^{(i)} - x_j^{(i)}\|^{p_j}\right) = 1 \tag{6.26}$$

$$\mathbf{1}^T\boldsymbol{\Psi}^{-1}\boldsymbol{\psi} = \mathbf{1}^T\mathbf{1} = 1 \tag{6.27}$$

Substituting Equations 6.26 and 6.27 into Equation 6.25, yields $\widehat{s}^2(\boldsymbol{x}) = 0$. This follows the intuition that if we are interpolating a point at which we know the answer, the error in the prediction must be zero.

Furthermore, it is convenient to work with the square root of the the mean square error $\widehat{s} = \sqrt{\widehat{s}^2(\boldsymbol{x})}$. This provides a standard error for measuring uncertainties in our prediction [46]. This way, a 95% Confidence Interval (CI) can be defined as $(\pm 2\,\widehat{s}\,)$ as described by Koehler and Owen (1996) [49].

6.3.2 KG with nugget effect

When noisy observations are considered, a diagonal matrix must be added to the covariance matrix $\boldsymbol{C}$:

$$\boldsymbol{C}_\Delta = \boldsymbol{C} + \boldsymbol{\Delta} \tag{6.28}$$

where $\boldsymbol{\Delta} = \mathrm{diag}[\mathrm{var}(\varepsilon_1), \mathrm{var}(\varepsilon_2), \cdots, \mathrm{var}(\varepsilon_n)]$ contains the variances of the observations. Kriging method is thus only modified by using $\boldsymbol{C}_\Delta$ instead of $\boldsymbol{C}$. Figure 6.6 shows a Kriging model based on noisy observations. Each observation has a different noise variance. CI is the 95% Confidence Interval [70].

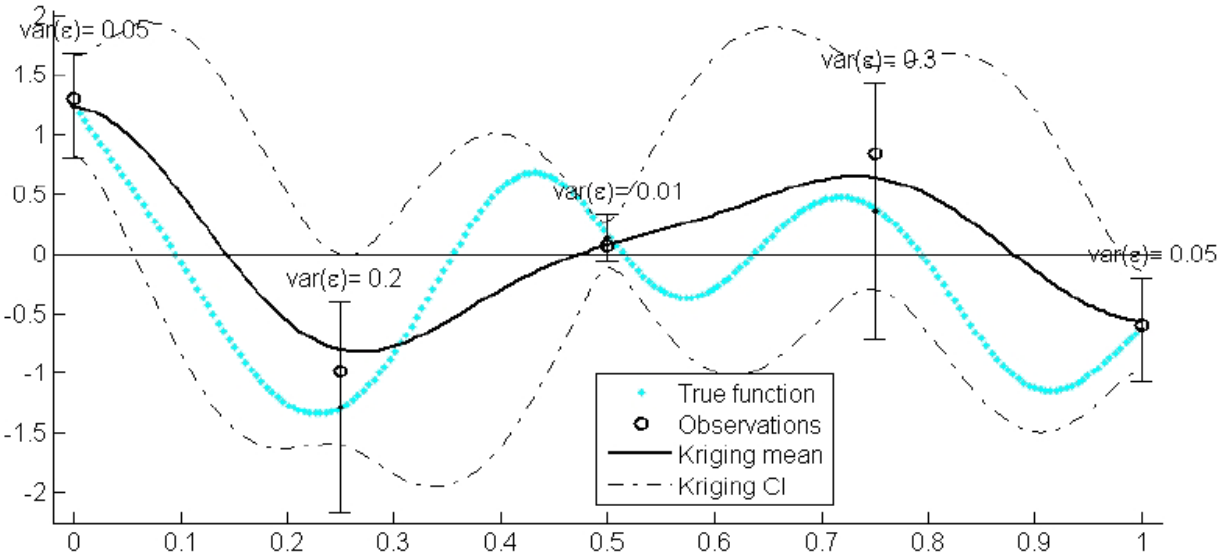


Figure 6.6: Example of Simple Kriging model with noisy observations. The bars represent ± two times the standard deviation of the noise. The Kriging mean does not interpolate the data, and the Kriging variance is non-null at the observation points. From Picheny, 2009 [70].

Among the advantages of the model, we count the fact that Kriging is the right choice for high-dimensional problems [43], and it can deal with applications that have small sampling data available. With the *Co-KG* variant, Kriging model can also deal with high and low fidelity input data. Simpson et al. (2001) [86] underline that if the function to be modelled is deterministic and highly non-linear in a moderate number of factors (less than 50), then KG may be the best choice.

However, Kriging has limited applicability when the number of design sites is large, mostly because determining the correlation parameters through the maximum likelihood estimation methodology can become computationally demanding [74] [44]. Over-fitting may also occur, and the risk is higher when there are very few design sites relative to the number of parameters to be tuned [74]. Finally, the correlation matrix can become singular if multiple sample points are spaced close to one another or if the sample points are generated from particular designs [44].

Table 6.4: Pros and cons of Kriging

Pros	Cons
– Can deal applications with small sampling data	– Sensitive to noise
– Can deal with highly non-linear problems	– Risk of ill-conditioned correlation matrix
– Flexible for different problems and large data	– High computational time

6.3.3 KG in DAKOTA

Currently, the Gaussian correlation function is the only option for the Kriging version included in DAKOTA. The form of the Gaussian Process is therefore the one presented in equation 6.23. The terms in the correlation vector and matrix are computed using a Gaussian correlation function and are dependent only on an k-dimensional vector of correlation parameters, $\boldsymbol{\theta} = \{\theta_1, \theta_2, \cdots, \theta_k\}^T$. By default, DAKOTA determines the value of $\boldsymbol{\theta}$ using a Maximum Likelihood Estimation (MLE) procedure. However, the user can also opt to manually set them by specifying a vector of correlation lengths $\boldsymbol{l} = \{l_1, l_2, \cdots, l_k\}^T$, where $\theta_i = 1/(2\, l_i^2)$ [5].

Six non-compulsory options can be used in Dakota to modify the characteristics of KG [3]. They are:

Correction factors, see Section 6.1.1.

Trend functions, whose purpose is to capture large-scale variations. Currently, only polynomials are supported (constant, linear, reduced quadratic, and quadratic). The reduced quadratic trend function includes the main effects, but not mixed/interaction terms.

Optimisation_method that changes the method used to find the optimal values of the hyper-parameters governing the trend and correlation functions. By default, the global optimisation method

DIRECT is used for Maximum Likelihood Estimation (MLE), but other options for the optimisation method are available.

Max_trials that is the maximum number of likelihood function evaluations.

Correlation_lengths that are usually optimised by Surfpack, however, the user can specify the lengths for the number of input dimensions manually.

Nugget/Find_nugget where one option has to be chosen. The former specifies a nugget to handle ill-conditioning, while the latter have Dakota automatically compute a nugget.

6.4 Meta-models summary

Table 6.5 represent a summary of the pros and cons of the three surrogate models here considered. The table is the result of data gathered through different reference sources. It will be later on compared with results obtained with DAKOTA toolkit.

Table 6.5: Summary of pros and cons and DAKOTA capabilities for each meta-model

Model	Pros	Cons	DAKOTA
KG	– Flexible – Good with highly non-linear problems – Good with high dimension problems	– Sensitive to noise – Risk of ill-conditioned correlation matrix – High computational time	– User can specify a way to handle ill-conditioning – User can speed up process by lowering down the number of optimisation iterations – DAKOTA User's Manual is very exhaustive on this model
RBF	– Ok with highly non-linear problems, but with enough sampling points – Closed form solution – Cheap to compute	– Requires high number of sampling points when dealing with highly non-linear problems – Sensitive to noise	– Not possible to change the basis function – User can modify the initial number of sampling points
ANN	– Ok with highly non-linear problems – Good with high dimension problems – Very flexible	– Large amount of data required to train it – User must have knowledge of how to select and prepare data	– SLP ANN can work with a smaller number of sampling points – User can change maximum number of hidden nodes and range of random weights

Analysis and comparison 7

In this chapter, the goal is to select the best combination of methods to build a Surrogate Based Optimisation that best suits the application required. The starting point is Table 3.7. The different methods, models, and algorithms are going to be evaluated with the use of test functions and the metrics described in section Section 3.4.2; in particular the Root Mean Square Error and the use of the Leave One Out technique.

7.1 Test functions

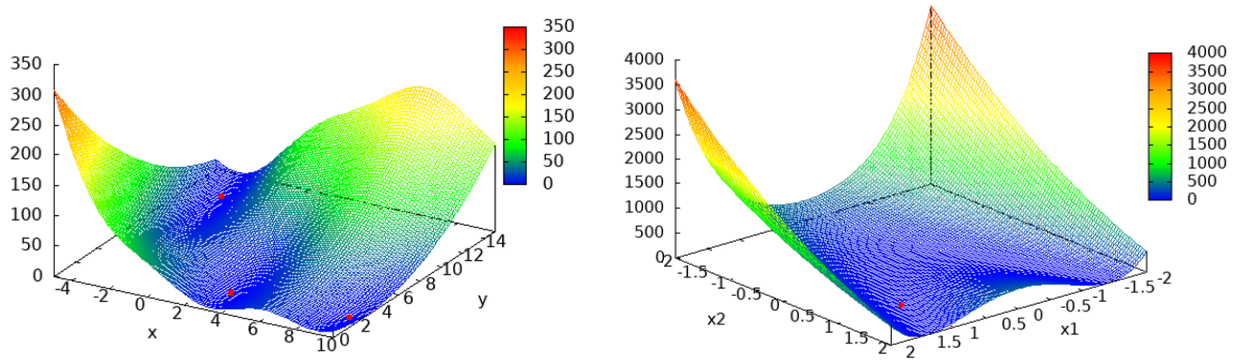
Table 7.1: Summary of test functions

FUNCTION	Variables	Boundaries	Objective	Optimum point(s)
Branin non-linear multi-modal	2	$-5 \leq x \leq 10$ $0 \leq y \leq 15$	$f \in [0, 350]$	$f(-\pi, 12.275) = 0.398$ $f(\pi, 2.275) = 0.398$ $f(9.425, 2.475) = 0.398$
Rosenbrock non-convex valley-shaped	2	$-2 \leq x, y \leq 2$	$f \in [0, 4000]$	$f(1, 1) = 0$
Hartmann3 non-linear	3	$0 \leq x, y, z \leq 1$	$f \in [-4, 0]$	$f(0.115, 0.556, 0.852) = -3.863$
1-D	1	$0 \leq x \leq 1$	$f \in [-7, 16]$	$f(0.757) = -6.021$

We already established in Chapter 3 that tests will have to be performed to compare different methods or algorithms. General rules when selecting a good test include [7]:

- Include many problems: the more problems that a test set contains, the more reliable the results of the experiment are. Here three functions will be used;
- Represent the application. Algorithms that perform very poorly for certain styles of problems may perform very well for other problems. Whenever possible, select a test set that is representative of the end application;
- Avoid biased initial conditions. All algorithms should be started using the same information;
- Avoid hidden structures. Carefully examine test sets to ensure that no hidden structure is helping or hindering certain algorithms.

The following functions, summarised in Table 7.1, are used:



(a) *Branin function. In red the three points of minimum*

(b) *Rosenbrock function. In red the point of minimum*

Figure 7.1: Analytical functions used for testing surrogate models

Branin is a non-linear and multi-modal function. The function can be seen in figure 7.1a and it is defined in equation 7.1.

$$f(x,y) = \left(y - \frac{5.1}{4\pi^2}x^2 + \frac{5}{\pi}x - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos(x) + 10 \tag{7.1}$$

And it is subject to

$$\begin{cases} -5 \leq x \leq 10 \\ \\ 0 \leq y \leq 15 \end{cases} \tag{7.2}$$

The equation, with the given boundary conditions, has three minima. They are:

$$f\Big((-\pi, 12.275),\ (\pi, 2.275),\ (9.425, 2.475)\Big) = 0.398 \tag{7.3}$$

Rosenbrock is a non-convex valley-shaped function (Rosenbrock, 1960) [75]. The function can be seen in figure 7.1b and it is defined in equation 7.4.

$$f(x,y) = (1-x)^2 + 100\,(y - x^2)^2 \tag{7.4}$$

And it is subject to

$$-2 \leq x, y \leq 2 \tag{7.5}$$

The equation, with the given boundary conditions, has one minimum:

$$f(1,1) = 0 \tag{7.6}$$

Hartman 3 is a non linear 3 variables function. The function is defined in equation 7.7.

$$f(x_n) = -\sum_{i=1}^{4} \alpha_i \exp\left(-\sum_{j=1}^{3} A_{ij}(x_j - P_{ij})^2\right) \tag{7.7}$$

And it is subject to

$$0 \leq x_n \leq 1 \quad \text{for } n = 1,2,3 \tag{7.8}$$

With

$$\alpha = \begin{pmatrix} 1.0 & 1.2 & 3.0 & 3.2 \end{pmatrix}^T \tag{7.9}$$

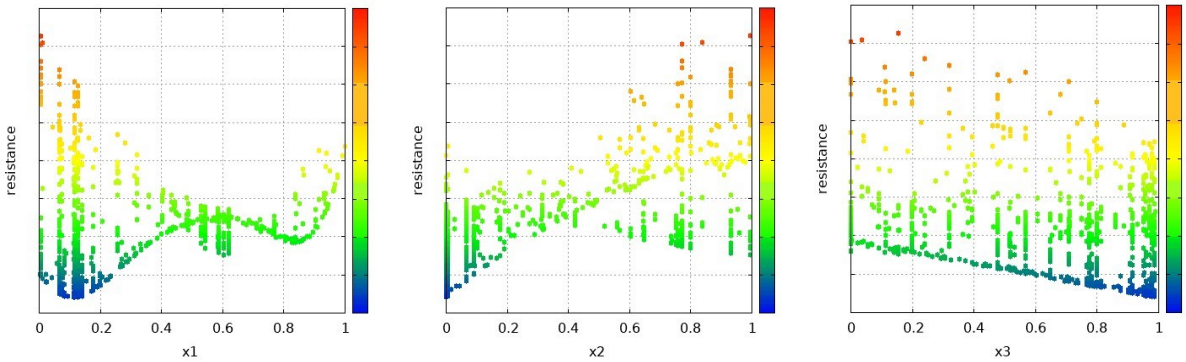
$$\boldsymbol{A} = \begin{pmatrix} 3.0 & 10 & 30 \\ 0.1 & 10 & 35 \\ 3.0 & 10 & 30 \\ 0.1 & 10 & 35 \end{pmatrix} \tag{7.10}$$

$$\boldsymbol{P} = 10^4 \begin{pmatrix} 3689 & 1170 & 2673 \\ 4699 & 4387 & 7470 \\ 1091 & 8372 & 5547 \\ 381 & 5743 & 8828 \end{pmatrix} \tag{7.11}$$

The equation, with the given boundary conditions, has four local minima and one global maximum:

$$f(0.114614, 0.555649, 0.852547) = -3.86278 \tag{7.12}$$

The Branin function is chosen because it can easily represent the output of a CFD study [28]. It is also quite common when testing the performance of different meta-models [22] [95] [47] [70] [46] [45] [1]. The Rosenbrock function is largely used as tests for optimisation problems, examples can be found in [5] [16] [88] [95] [1]. Finally, the Hartmann 3 function has been chosen because it represents a commonly used test with 3 variables [95] [47] [1]. In addition to these three functions, in Section 5.2 a 1-D function $f(x) = (6x-2)^2 \sin(12x-4)$ was used to illustrate the functioning of infill plans. It is frequently used for that purpose, as we can see from [23] [22] [73] [39] [40].



(a) *Surrogate of Rt over x_1 = bow width* **(b)** *Surrogate of Rt over x_2 = x shoulder position* **(c)** *Surrogate of Rt over x_3 = bilge radius*

Figure 7.2: Example of surrogate surfaces of resistance for a bow ship optimisation performed at Van Oossanen N.A.

We report in Figure 7.2 an example of surrogate surface for a bow ship optimisation performed at Van Oossanen N.A. to give an idea of what can be encountered. The test functions were chosen analysing several old optimisations such the one here presented.

7.2 Sampling plan comparison

The first decision to make is which sampling plan to use. The two best candidate solutions selected and described in Section 5.1 are the Halton sequence and the Latin Hypercube algorithm. These two methods are compared with the test functions described in Section 7.1 and the use of DAKOTA toolkit. In order to only analyse the influence of Halton and Latin Hypercube algorithms on the surrogate models generation, the number of sampling points was kept fixed. Also, as LHS is a random process, the tests for this algorithm are repeated five times, and the numbers shown here are the mean.

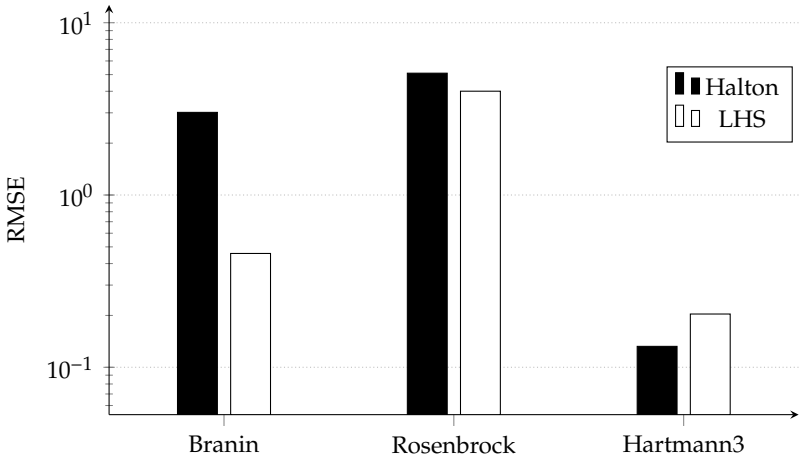


Figure 7.3: Comparison of Halton and LHS sampling plans for KG with the use of three analytical functions

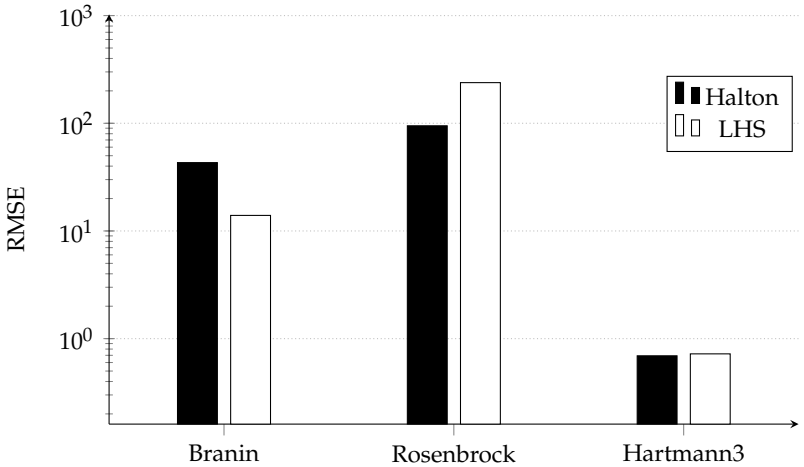


Figure 7.4: Comparison of Halton and LHS sampling plans for ANN with the use of three analytical functions

The results are plotted in terms of RMSE [13] (see Section 3.4.2) in Figure 7.3 for the Kriging model, in Figure 7.4 for Artificial Neural Network, and in Figure 7.5 for Radial Basis Function. Note that they are in a logarithmic scale.

What the graphs show is that Halton sequence and LHS sampling plans look quite equivalent. One does not always perform better than the other. Therefore the choice should depend, for example, on user-friendliness or infill sampling plan requirements. If no reasons suggest otherwise, Halton is preferred as it is the one already used at Van Oossanen N.A.; thus, the company members are already familiar with

13: The RMSE errors here presented are obtained after a Leave-One-Out analysis. To understand why this numbers are used, see Section 7.3 and [5].

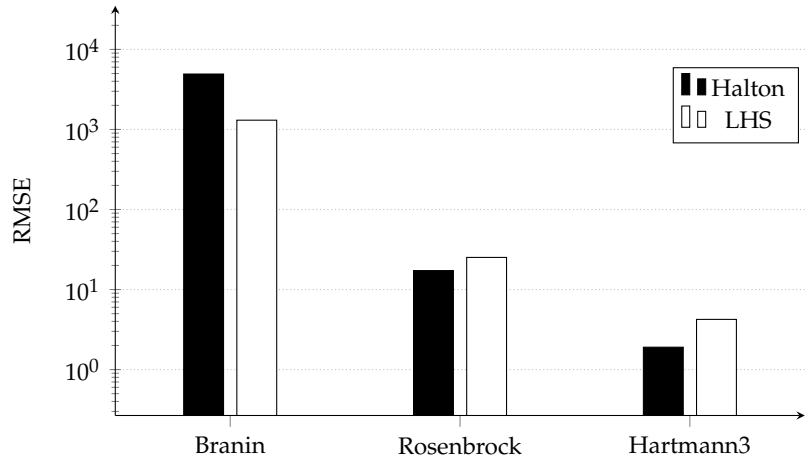


Figure 7.5: Comparison of Halton and LHS sampling plans for RBF with the use of three analytical functions

it. Also, it has no random component, so that it is easier to spot and correct possible mistakes.

7.2.1 Number of samples

A second important question to answer is the number of total sampling points to use. When constructing a surrogate, this factor has been seen to have a big influence on the quality of a meta-model (Section 3.3.1 and Section 3.4.2). For this test, to avoid biased conditions, only the Halton sequence was used. Each surrogate model was generated with $8d - 20d$ points. d is the number of design variables, 2 in the case of Branin and Rosenbrock functions and 3 in the case of Hartmann3 function.

The results are plotted in terms of RMSE [14] in Figure 7.6 for Kriging, in Figure 7.7 for ANN, and in Figure 7.8 for RBF. Note, they are in a logarithmic scale.

14: The RMSE errors here presented are obtained after a Leave-One-Out analysis. To understand why this numbers are used, see Section 7.3 and [5].

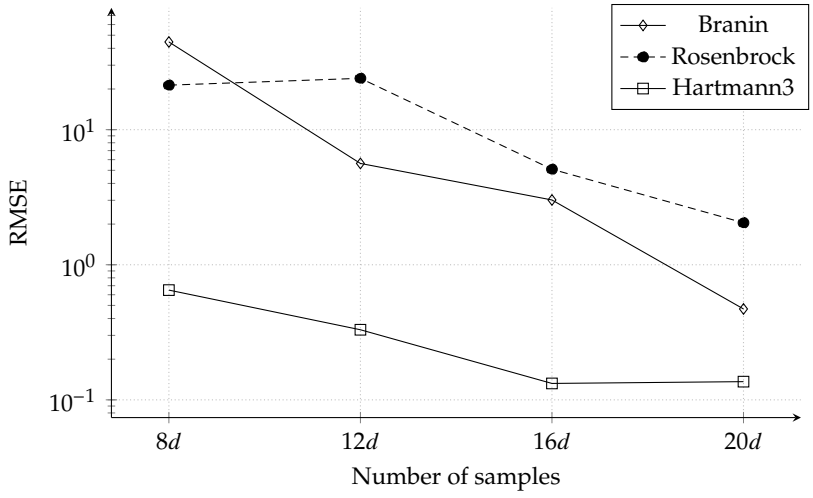


Figure 7.6: Comparison of KG quality when using different number of sampling points

It looks like, as we predicted, that the error tends to lower down as the number of sample points rises. In particular, at around $16d$, the metrics almost stabilise for most of the cases. This would mean that for three-dimensional analysis, a total of 48 simulations should

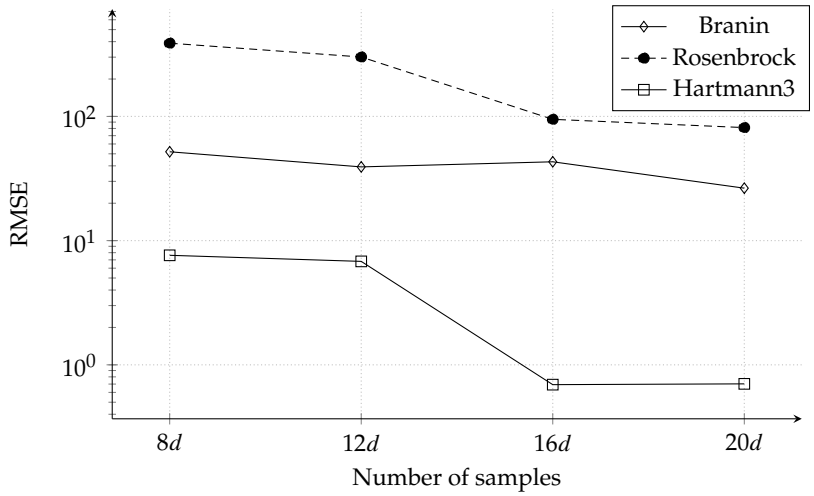


Figure 7.7: Comparison of ANN quality when using different number of sampling points

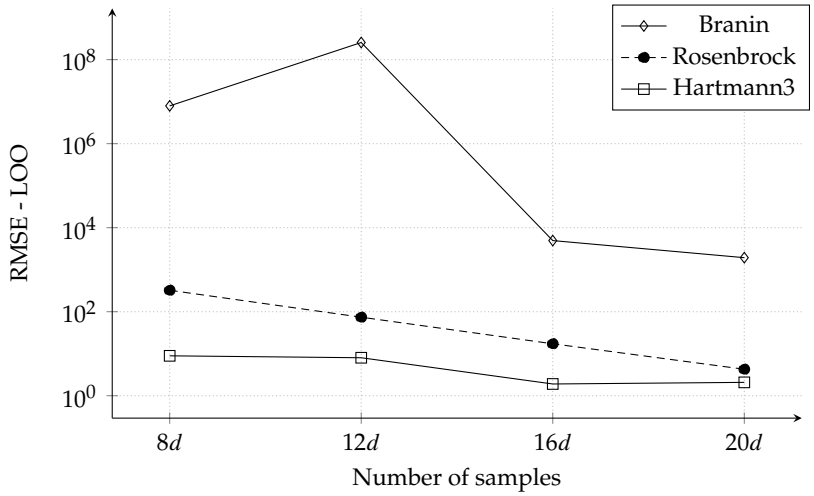


Figure 7.8: Comparison of RBF quality when using different number of sampling points

be performed. Currently, at Van Oossanen N.A., an optimisation with three variables uses only 30 points (10d). Therefore, in order to improve the quality of the routine, the number of (total, initial and refinement) sampling points is slightly increased. Per the company's budget, the number of points is defined to be in the range of 11d - 15d.

7.3 Meta-model comparison

Now the three meta-models described in section Section **??** are tested. In particular, to obtain a fair comparison between the results of different functions and different models, for each computation, Halton sampling plan was used. The number of sampling points was fixed to 16d. Moreover, no corrections or tuning are applied to the meta-models, and DAKOTA default settings are used. This has the purpose of looking for the best fitting model as well as the easiest-friendly one.

The results are displayed in table 7.2 and plotted in figure 7.9 in terms of RMSE with a log scale. What we can see is that the Radial Basis Function seems to perform best for Branin and Rosenbrock.

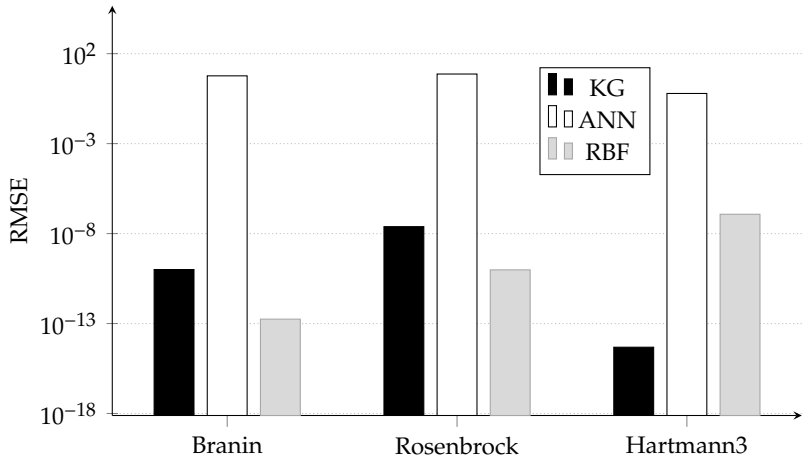


Figure 7.9: Comparison of KG, ANN, RBF metrics for three different analytical functions with no corrections

However, not for Hartmann3, where KG has a lower error. On the other hand, Artificial Neural Network always seems the worst when trying to emulate the surfaces of the test functions. However, as also stated in the DAKOTA User manual [5], caution is advised when applying and interpreting these metrics. In general, lower errors are better, but for interpolatory models like Kriging models and Radial Basis Function, it will almost always be zero. Leave-One-Out Cross-Validation analysis will provide more reliable estimates of the true model prediction error.

Model	RMSE	MaxAE	CPU time [s]
BRANIN			
KG	1.00E-10	2.36E-10	1.47
ANN	5.84E+00	1.22E+01	0.16
RBF	1.76E-13	5.58E-13	1.82
ROSENBROCK			
KG	2.43E-08	5.14E-08	1.17
ANN	7.36E+00	2.65E+01	0.17
RBF	9.61E-11	1.90E-10	1.80
HARTMANN3			
KG	4.84E-15	1.78E-14	2.46
ANN	6.11E-01	2.22E+00	0.49
RBF	1.18E-07	3.38E-07	5.95

Table 7.2: Summary of DAKOTA metrics and CPU time in seconds for each meta-model applied to three different functions with no corrections

Table 7.2 also shows the running time of each method in seconds. KG is demonstrated to be the slowest for two cases out of three, exactly what the analysis carried out in Section 6.3 pointed out. However, this will not be an essential parameter in the choice of which meta-model to use, because even the highest time (6 seconds) is nothing compared to one CFD simulations (20 hours).

7.3.1 Leave-One-Out analysis

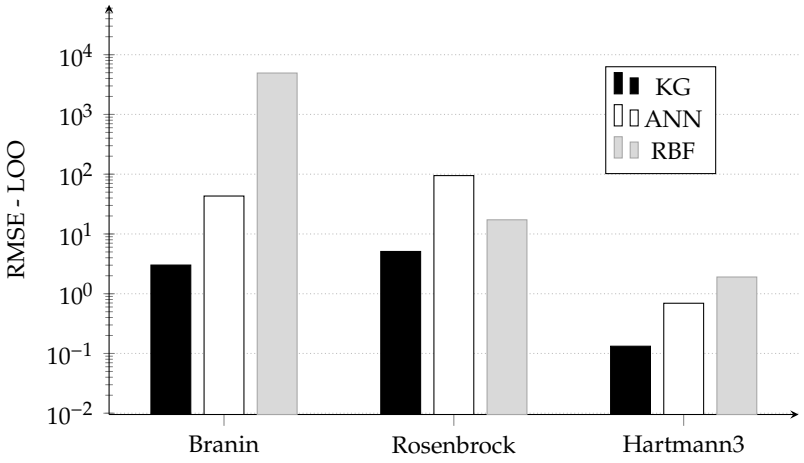


Figure 7.10: Comparison of KG, ANN, RBF Leave One Out metrics for three different analytical functions with no correction

The same tests performed in Section 7.3 are now shown with the same settings, but after a Leave One Out analysis is performed by DAKOTA (see Section 3.4.2).

Table 7.3: Summary of DAKOTA Leave-One-Out metrics for each meta-model applied to three different functions with no correction

Model	RMSE	MaxAE
BRANIN		
KG-LOO	3.02E+00	1.68E+01
ANN-LOO	4.31E+01	1.72E+02
RBF-LOO	4.93E+03	2.78E+04
ROSENBROCK		
KG-LOO	5.09E+00	1.94E+01
ANN-LOO	9.48E+01	2.56E+02
RBF-LOO	1.72E+01	5.00E+01
HARTMANN3		
KG-LOO	1.32E-01	3.84E-01
ANN-LOO	6.93E-01	2.95E+00
RBF-LOO	1.90E+00	6.42E+00

The results are displayed in Table 7.3 and plotted in Figure 7.10 in terms of RMSE-LOO and with a log scale. What the graph shows is that, in contrast with what was presented in Figure 7.9, it is KG method that overall has the best performance for every test function. It was not possible to give a reasonable explanation for why the RBF method performs so poorly compared to the other two. Considering what the literature review pointed out on the matter in Section 3.4, a reason could be that not enough sample points were selected for such highly non-linear test functions. However, this was not further investigated, because, in the SBO routine constructed for this thesis, the number of points cannot be higher than the $16d$ used for this test.

To conclude, the desire is to implement the Kriging model. Also, on top of this DAKOTA analytical tests, we remember the advantages

of the method pointed out in Section 6.3. In particular, KG was found to be suitable for highly non-linear problems, it is an interpolation method thus the solution on true points is exact, it is flexible, and DAKOTA's manuals are very exhaustive on this model. Furthermore, the choice of the Kriging method allows us also to select also the Expected Improvement function for the infill sampling plan. However, before finalising the choice, the concern rises because among its drawbacks KG counted a high sensitivity to noise.

7.4 Noise influence

In Section 2.1.1, the issue of noise and its effects on surrogate models was raised. Furthermore, in Chapter 6, we saw that different methods deal with noisy data differently.

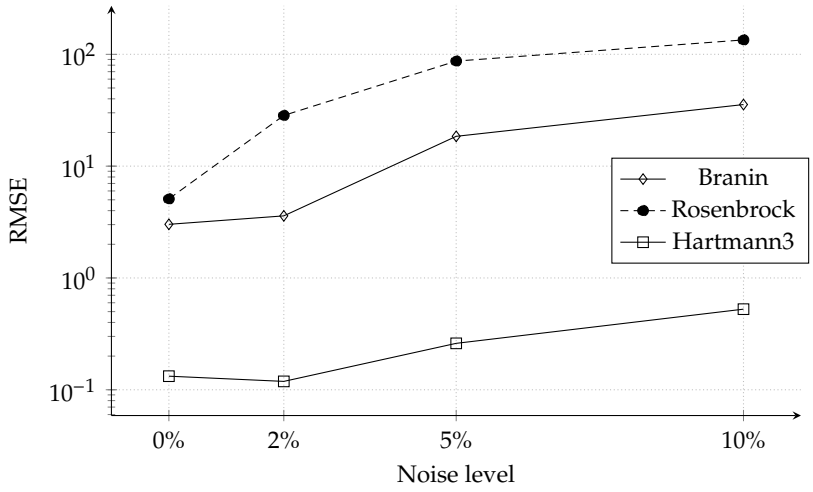


Figure 7.11: KG metrics with different levels of noise on training data for four different analytical functions

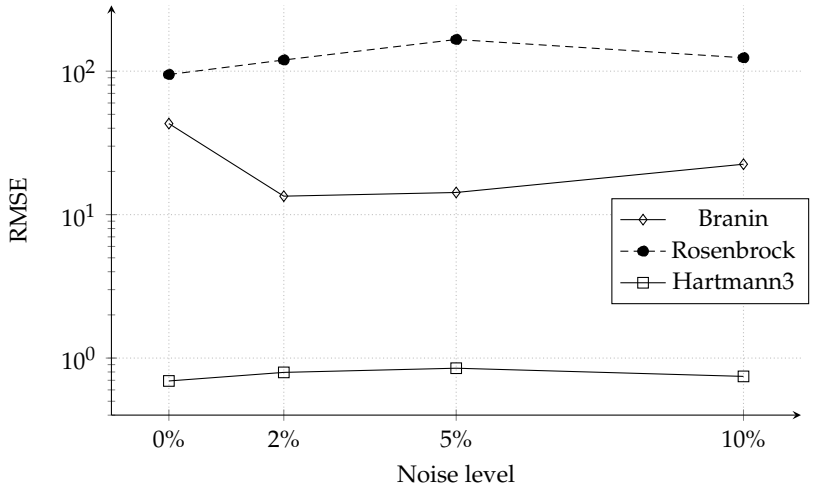


Figure 7.12: ANN metrics with different levels of noise on training data for four different analytical functions

The three test functions described in Section 7.1 are now tested with noise. However, as sample data created analytically does not have any noise, in order to consider its influence artificial noise is added to the response values y as:

$$\widehat{y} = f(\boldsymbol{x}) \cdot (1 - l\,\delta) \tag{7.13}$$

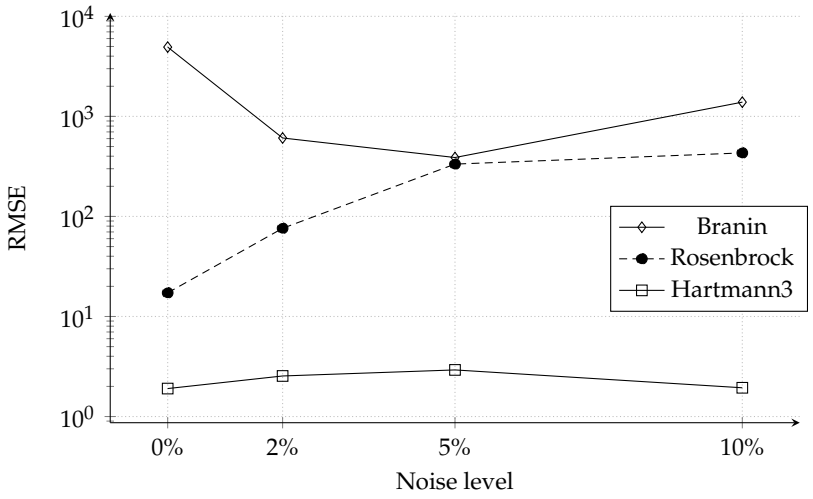


Figure 7.13: RBF metrics with different levels of noise on training data for four different analytical functions

where $l = 0\% \sim 10\%$ is a scaling parameter and δ is random number sampled from the standard Gaussian distribution $N \sim (0, 1)$.

In particular, to start each model with information as similar as possible, Halton sampling plan was chosen, and the number of sampling points was fixed to $16d$. Four levels of artificial noises, $0\%, 2\%, 5\%, 10\%$ are added to the response values of the corresponding sample points using equation 7.13. We remember that the expectation is to have a noise of around 5% when performing a CFD simulation.

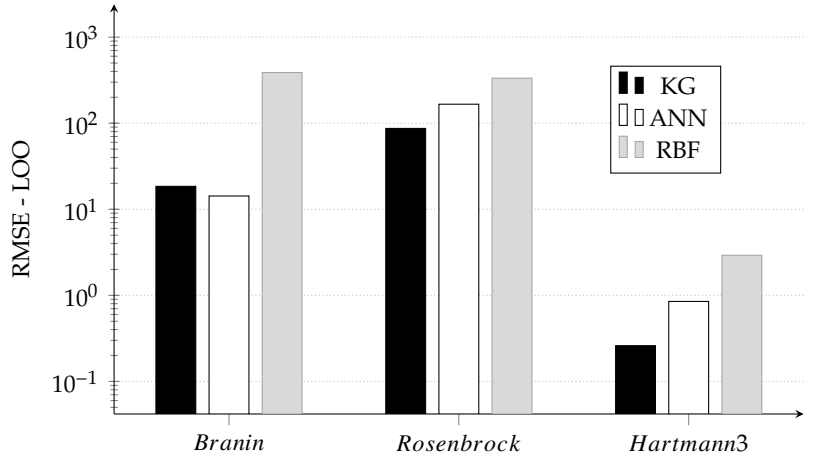


Figure 7.14: Comparison of KG, ANN, RBF Leave One Out metrics with 5% level of noise for four different analytical functions

The results are shown in Figure 7.11 for KG, in Figure 7.12 for ANN, and in Figure 7.13 for RBF. They show the RMSE on a logarithmic scale for increasing level of noise. We can see that, as we predicted in Section 6.3, KG is the most sensitive to noise. The method interpolates between the points exactly and thus include any possible noise in the surface fit. The error of the method rises when the noise level also rises. Also, ANN is not influenced by noise, as the method tends to smooth the response function. Finally, figure 9.16d shows that when noise does not exceed 5%, KG method still gives a better solution than ANN. To conclude, the Kriging method is chosen.

7.5 Optimisation algorithms

The next decision to make is which optimisation algorithm to use. In Section 3.2, the literature review pointed out how most authors focused on Genetic Algorithm, and also this is the best derivative-free global method supported in DAKOTA (Table 3.3). Nonetheless, we recognised also the value in paring a global method to a local method. In Chapter 4, two possible methods were analysed: Pattern search and Quasi-Newton BFGS.

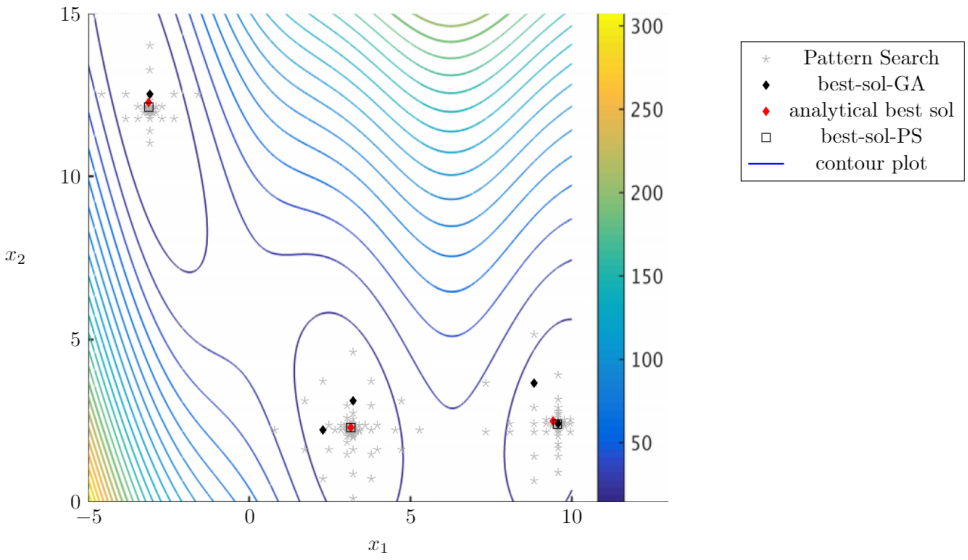


Figure 7.15: Trajectory plot (grey stars) for a Pattern Search local algorithm applied to a KG surrogate surface representing a Branin function (contour lines). In red the analytical best solutions, in black the best solutions found by the global GA search, in squares the best solutions found by PS.

Figures 7.15 and 7.16 represent the application of PS and QN, respectively, on the Branin function represented by a contour plot. For both figures, a KG meta-model was constructed using the same 32 points part of the Halton sequence. Also, the settings of GA were the same (Table 4.2). The five best points found by GA, in black in the plots, are used as starting points for the local algorithms. In both cases, the three analytical optimum points are in red, while in grey we see the trajectories followed by PS or QN, and the rectangles show the three best solutions found by the local algorithms.

What these figures show, is that indeed the Genetic Algorithm is already quite precise in determining the optimum points. Mainly because the case of Branin function is particularly challenging to analyse, as it is multi-modal, moreover, all three best solutions were identified. However, we can also see that the use of a local algorithm can improve the prediction significantly. Furthermore, only the use of a local algorithm would not be advised because the location of the starting point greatly influences them. Thus, in conclusion, the use of two different methods in a sequence is confirmed to be the best and most balanced solution.

Another important conclusion that can be drawn from Figures 7.15 and 7.16, and was already anticipated in Section 3.2, is that since the

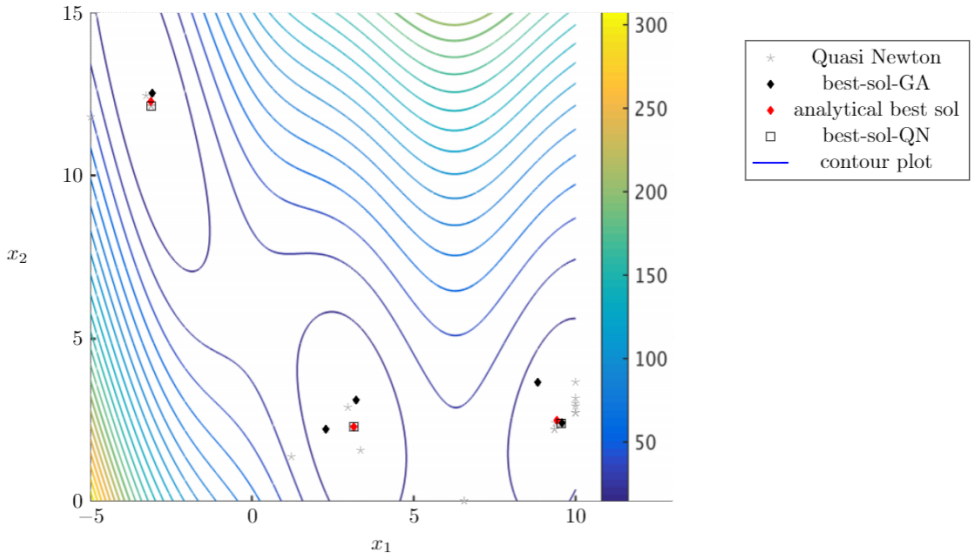


Figure 7.16: Trajectory plot (grey stars) for a Quasi-Newton local algorithm applied to a KG surrogate surface representing a Branin function (contour lines). In red the analytical best solutions, in black the best solutions found by the global GA search, in squares the best solutions found by QN.

optimisation is performed on the surrogate level, it does not matter which local algorithm is going to be used. Both PS and QN give roughly the same results with the same initial conditions. Thus, the decision of which one to use is not based on performance, but visualisation. Pattern Search is more transparent for the user to inspect than gradient-based algorithms, thanks to its fixed and straightforward pattern (see Section 4.3). If more than one solution is returned from PS and it is still not clear which one is best than the user can make use of the gradient information returned by DAKOTA. This way, a more robust solution will be preferred to a possibly unstable one.

7.6 Refinement points

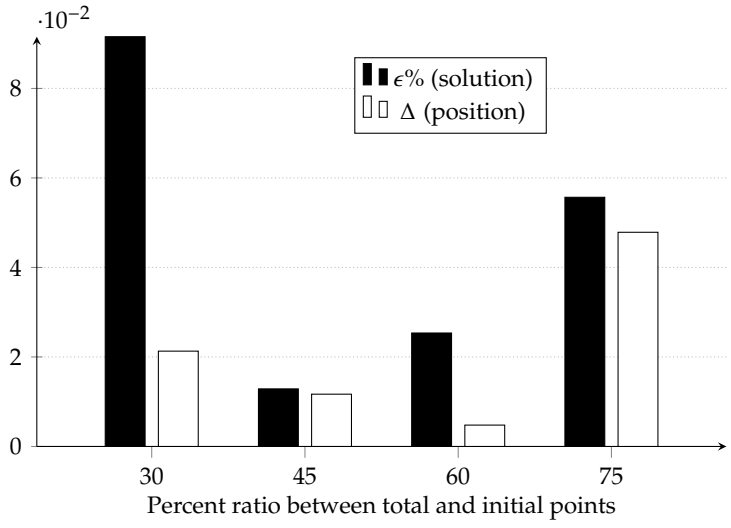


Figure 7.17: Comparison of EI performance when changing the percent ratio between total and initial points for Branin function[a]

[a] distances are with respect to one of the three optimum point location

The last decisions that are necessary to complete all the information

necessary to build an SBO routine are about the infill sampling plan. In Section 3.3.3, when we conducted the literature review on the topic, and in Section 5.2 when we analysed the algorithms of different infill techniques, appeared clear that the best one available in DAKOTA was EI. However, this method could only be used with a Gaussian process-based meta-model. Now that it is questionless that KG is going to be used, we can also confirm the use of EI.

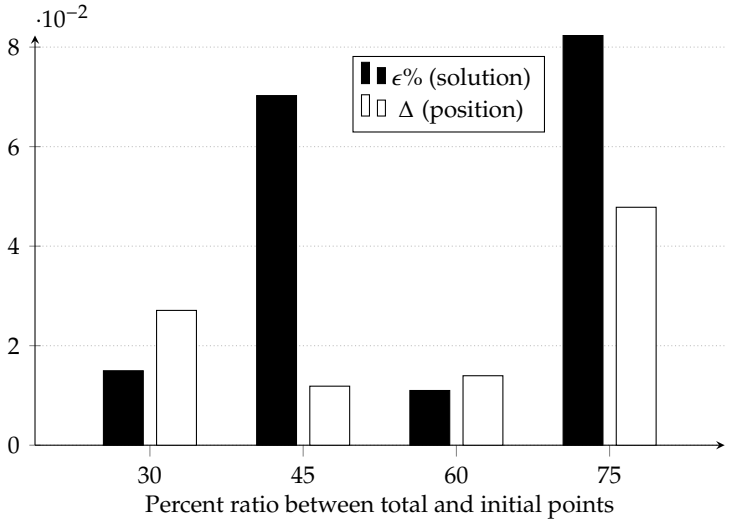


Figure 7.18: Comparison of EI performance when changing the percent ratio between total and initial points for Rosenbrock function.

The main issue that still needs to be solved is the ratio between total sample points available versus the number of initial points used to construct KG. In Section 3.2, we saw that according to Sobester et al. (2005) [88], the infill plan should contain between one-third to two-thirds of the total available points. Thus, the last test has the aim to decide a fixed percentage ratio between total and initial points.

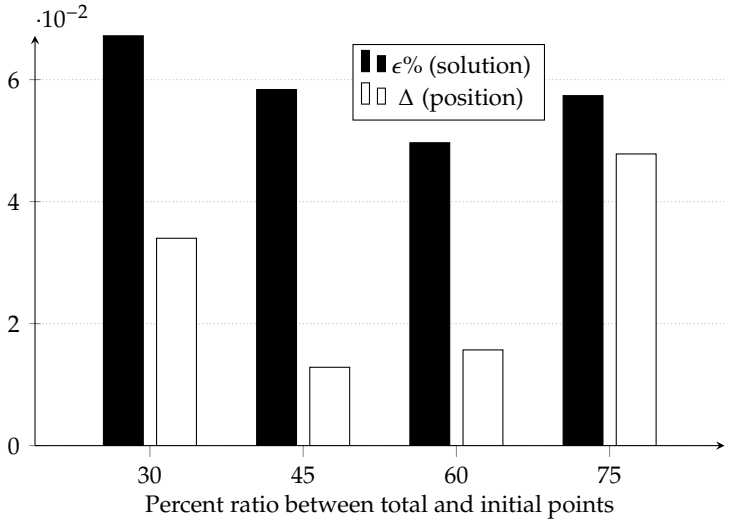


Figure 7.19: Comparison of EI performance when changing the percent ratio between total and initial points for Hartmann3 function.

Figure 7.17 for Branin function, Figure 7.18 for Rosenbrock function, and Figure 7.19 for Hartmann3 function show the performance of Expected Improvement function when 30%, 45%, 60%, 75% initial points out of the total available points are used to construct the KG meta-model. The total number of available points are, in accordance with analysis in Section 7.2.1, are 26 when $d = 2$ and 35 when $d = 3$

with d the number of variables. This way we stay within the $12d$ - $15d$ band that was selected. The initial sampling plan points are part of the Halton sequence.

To measure the performance of EI, we use two different values in the plots: ϵ% represents the percentage error of the solution found by EI for the optimal analytical solution. Note, that for the Branin function (Figure 7.17) this value is shown for one best solution, the one found by EI. On the other hand, Δ represents the Euclidean distance between the position of the solution found by EI and the location analytically known of the best solution.

What these graphs show is that indeed Expected Improvement function is extremely useful in refining the design space. This is because the errors ϵ% are all below 1 per cent. However, the best overall performance is found to be when the ratio between total and initial points is between 45% and 55%.

One final analysis deals with the convergence of EI function. Two examples of lousy convergence criteria are shown in Figure 7.20. To the left, we see that the maximum number of possible evaluations is too low, and the EI function has no possibility of exploring the design space. Both the local and the global minimum are missed. Whereas, to the right, we see that tolerance is too low and too many points are added at the same location (see points 9-12 in the bottom zoom graph), and no more useful information is added.

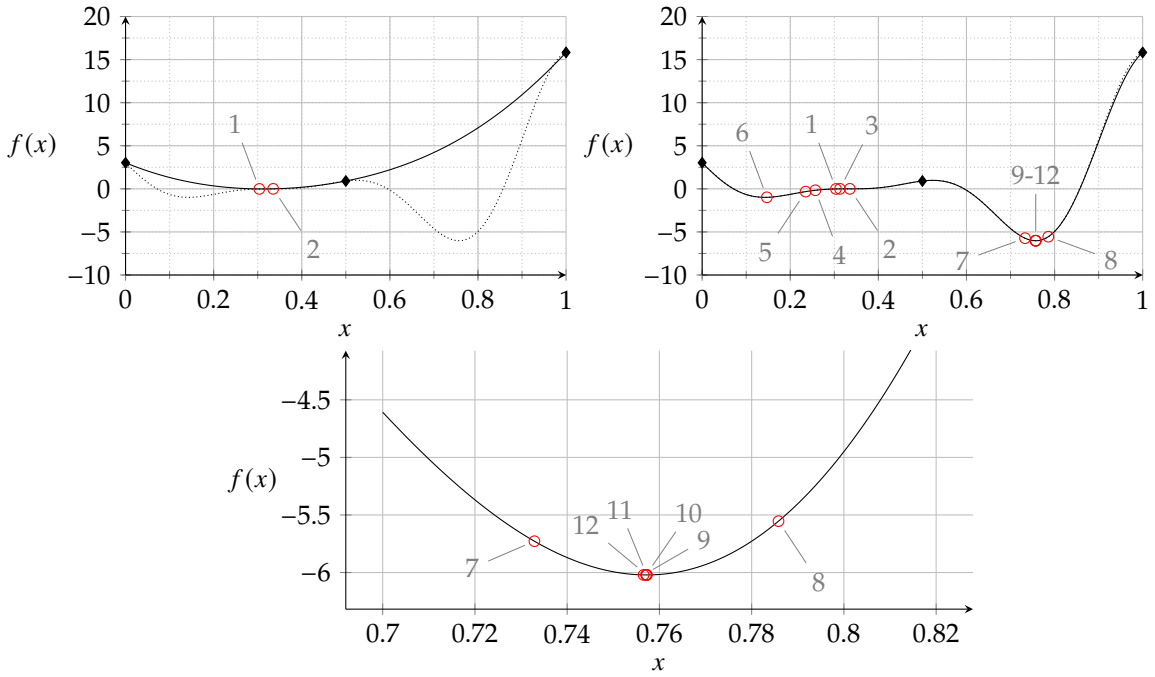


Figure 7.20: Two examples of bad EI convergence criteria. To the left: maximum number of possible evaluations is too low and EI function has no possibility of exploring the design space. Both the local and the global minimum are missed. To the right: tolerance is too low and too many points are added at the same location (see bottom graph) and no more useful information is added.

Unfortunately, nothing can be done to avoid the first situation (maxi iteration number too low), because the total amount of available points is fixed. Also, it is challenging to understand when this is happening, especially without adding new points. If the routine

should miss the global minimum, we expect to obtain a (less considerable) resistance reduction. Nevertheless, for the second situation (convergence tolerance too low), a x-convergence tolerance can be set. In this case, the Efficient Global Optimisation method will terminate if the relative change in the best decision variables values x is less than the specified tolerance. For more information, see Section 5.2.1 or DAKOTA's manuals [5] [3]. Only experience will tell what the best value to use is, but it better a smaller value that allows more certainty on global optimum than a higher one that kills the simulations too early. Thus, for now, a `x_con_tol` = 0.05 is set.

7.7 Conclusive remarks

Item	Adopted solution
Sampling	Initial sample method – Halton sequence
	Total sample size – Between 12d and 15d [a]
Surrogate	Kriging
Optimisation	Combination of global and local algorithms – Global: GA – Local: PS
Infill	Expected Improvement
	Percentage of initial points over the total – Between 45% and 55%
	Convergence criteria – Maximum number of evaluations reached – $\boldsymbol{x}$-convergence = 0.05 reached

Table 7.4: Summary of conclusions drawn from analysis and comparison of test functions

[a] d is the number of variables

In conclusion, as one of our aims was to give some guidelines to follow when building up an SBO routine with CFD expensive simulations, the recommendation is to start by carefully analysing the objective functions that will have to be approximated. This analysis should end with the selection of test functions that best and fully represent different possible scenarios. The reason for this is that most of the decisions made were based on the results obtained on those functions. Afterwards, the suggestion is to follow the order of Chapter 7 to make decisions and select appropriate methods and algorithms.

Table 7.4 summarises all the conclusion drawn specifically for the context of this thesis.

Design applications

Validation test: Hull Vane

8

The first test with real CFD data representing the resistance of a hull shape is a Hull Vane optimisation case already successfully solved at Van Oossanen N.A.. In the original case, Van Oossanen N.A. was asked to investigate the use of energy-saving hull appendages for a ferry. A Hull Vane was thus added to the benchmark hull. The geometry of the hull (side, bottom, aft, and forward views) with the addition of a Hull Vane at the stern of the ship is visible in Figure 8.1.

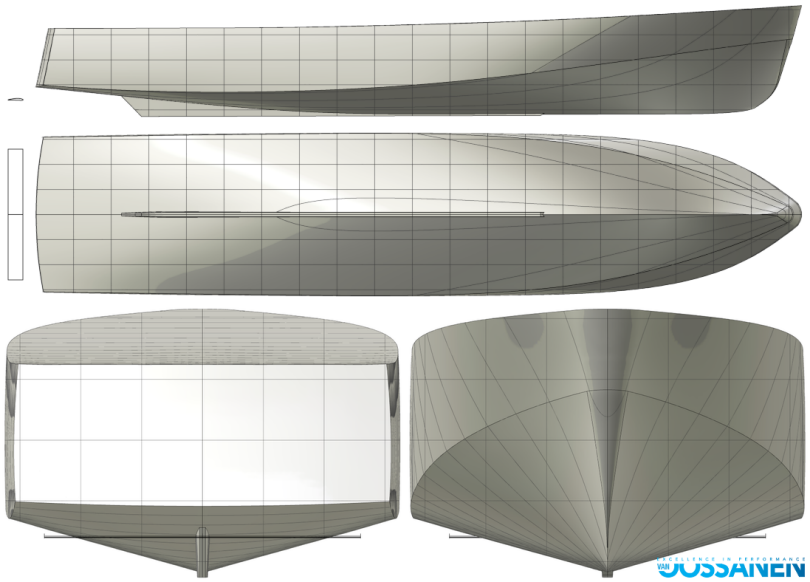


Figure 8.1: Geometry of the 'Valais' ferry with a Hull Vane

The optimisation performed by Van Oossanen N.A. showed how the additional appendage gave a reduction in total resistance for the benchmark geometry. Furthermore, a specific configuration of Hull Vane parameters was found to be particularly advantageous: a reduction of 15.4% was achieved.

This test is going to be performed with the same simulation settings, the same benchmark geometry, the same variables and boundaries, and the same modeller for geometry modifications as the original one. Only the SBO settings were changed. Thus the routine built with the choices made in Chapter 7 is considered validated if the results are comparable to (or better than) the original ones. The comparison is based both on the coordinates of the variables representing the best point in the design space as well as on the resistance reduction for the benchmark hull. All the information regarding the case is a courtesy of the company, and it is part of their official reports. Due to intellectual property of Van Oossanen N.A. on the data, all the values in this thesis are presented using non-dimensional numbers or percentages.

8.1 Hull vane

Pieter Van Oossanen invented the Hull Vane® (HV) in the early '90s. Its first application was on a catamaran, to reduce its excessive running trim and increase the vessel's top speed. Since then the Hull Vane's resistance and motion reduction properties have been studied and optimised. Currently, the Hull Vane is mainly applied to monohulls sailing at moderate to high non-planing speeds (Froude of 0.2-0.5).

According to Uithof et al. (2016) [91], the Hull Vane can reduce the resistance up to 25.5%, or increase up to 9.5%, depending on the hull form, speed, sea state, and running trim. Three different physical phenomena mainly cause the calm water resistance reduction:

- Transom wave reduction: The low-pressure region created by the Hull Vane, interacts with the wave crest created at the stern, reducing it.
- Thrust generation: The Hull Vane uses the up-wash at the ship's stern to create forward thrust.
- Trim correction: It is common in ships to have excessive trim aft due to the low-pressure region created by the rocker. In these cases, the Hull Vane also corrects the trim, reducing the resistance.

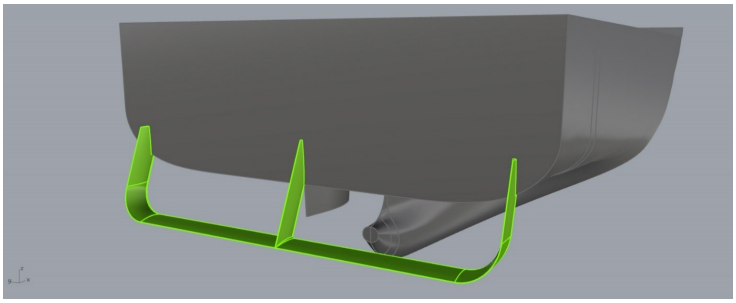


Figure 8.2: Example of Hull Vane

Figure 8.2 represents an example of a Hull Vane structure positioned at the stern of a hull.

8.2 Simulation Settings

All numerical simulations described in this thesis are performed with the software tools illustrated in Chapter 1. The purpose of the simulations is to obtain a resistance value for a particular hull geometry. The ship motions that are calculated by 6 degrees of freedom solver were restricted only to allow trim and heave besides the imposed velocity. The simulation is started at zero speed after which the ship is gradually accelerated to the final velocity, using a fourth of a sinusoidal ramp. To achieve a stable solution, 2000 time steps are used.

A "volume of fluid" method is used to account for the free surface (i.e. both water and air flows are solved), for which the parameters are given in Table 8.1. The solver uses the (Unsteady) Reynolds Averaged Navier Stokes (U)RANS equations which describe the flow. These

equations need a closure model for which the two-equation $k-\omega$ SST Menter turbulence model was used. The freestream turbulence quantities were initialised using the reference length and velocity. Wall functions were used to simulate the flow in regions very close to solid walls, reducing the mesh density requirements in the boundary layer.

Fluid properties			
Fresh water viscosity	μ_w	0.001138	Pa s
Water density	ρ_w	1000	kg m^3
Air viscosity	μ_a	0.0000185	Pa s
Water density	ρ_a	1.2	kg m^3

Table 8.1: Water and air parameters used for 'Valais' test case

8.2.1 Variables and constraints

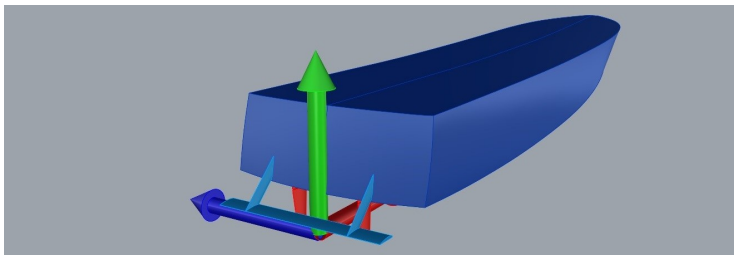


Figure 8.3: Hull coordinate system adopted in Van Oossanen simulations

The coordinate system adopted is the right-handed Cartesian coordinate system. The positive x-direction is defined as the direction from aft to forward (red arrow). The positive y-direction is defined as the direction to the port side of the vessel (blue arrow). The positive z-direction is defined in an upward direction(green arrow). It follows that negative trim is bow-up and positive rise indicates a positive vertical movement of the hull at the centre of gravity. The origin of the coordinate system is located at the baseline at the centre line aft of the transom (see Figure 8.3).

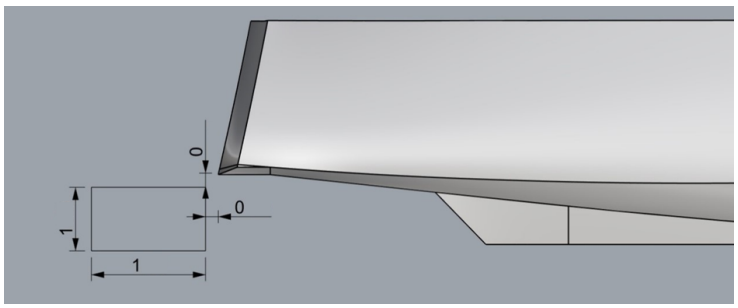


Figure 8.4: Boundaries in which the Hull Vane can be placed during optimisation

For this optimisation, the Hull Vane is free to move in the horizontal and vertical direction, within certain boundaries (see Figure 8.4). The Hull Vane chord length represents the third variable (see Figure 8.5).

The shape of the section does not change, and span is fixed to $0.8B_{WL}$, with B_{WL} the waterline beam.

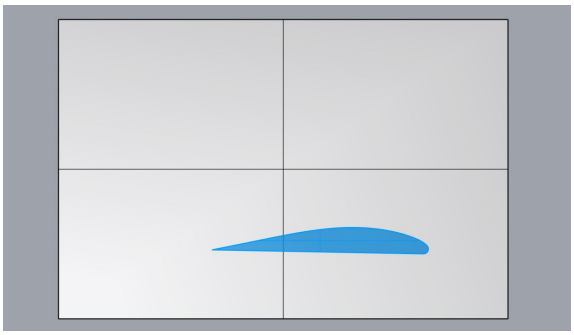


Figure 8.5: Hull Vane section used for optimisation

For a more transparent analysis and visualisation of data, the boundaries for each variable are normalised and can vary only between 0 and 1.

8.2.2 Domain and mesh

The computational domain is aligned with the reference system and is made up of a rectangular box in which half of the hull geometry was placed. The domain around the hull is constructed such that the boundaries are far enough away so as not to influence the results. In setting up the domain, the default guidelines used by Van Oossanen N.A. were used.

All hull surfaces had a no-slip boundary condition using wall functions to capture the boundary layer. In the symmetry plane a mirror boundary condition was applied, and on the top and the bottom of the domain the pressure was prescribed. All other domain faces have external/free-flow boundary conditions with a prescribed flow speed of $v = 0$ m/s.

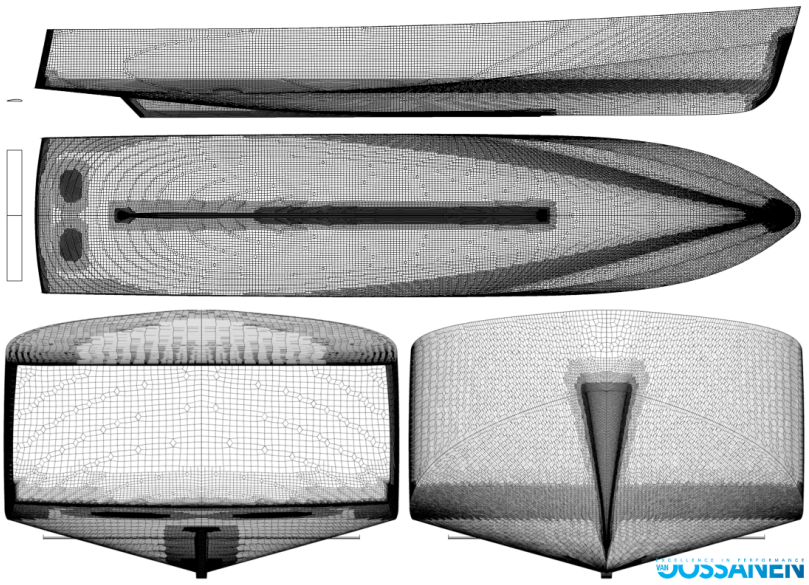


Figure 8.6: Example of mesh computed on the 'Valais' ferry

Example of the mesh on the surface of the hull is given in Figure 8.6.

8.3 SBO settings

Table 8.2: Summary of conclusions drawn from Chapter 7 for a $d = 3$ variable optimisation

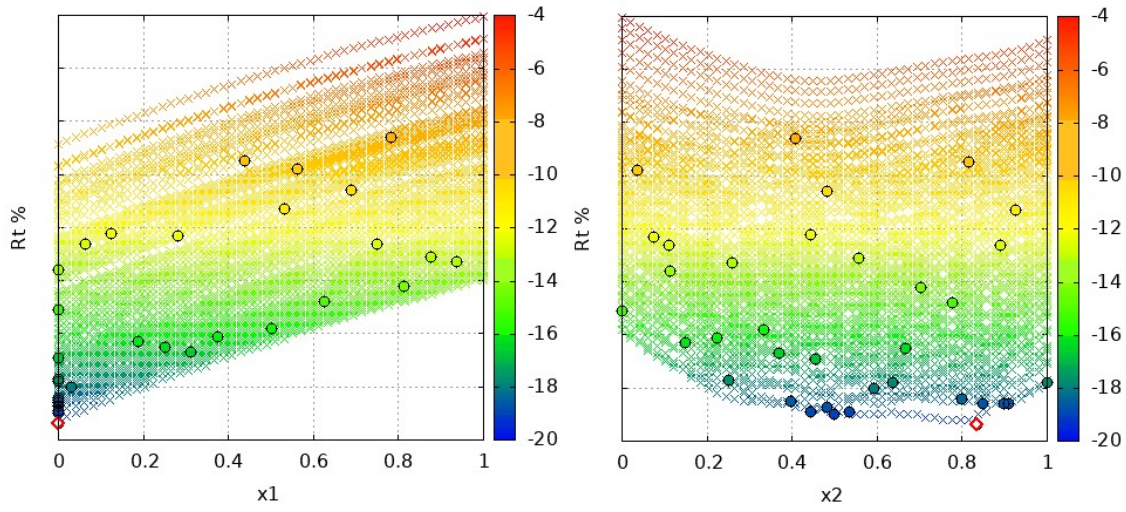
Item	Adopted solution
Sampling	Initial sample method – Halton sequence
	Total sample size – $35 \approx 12d$
Surrogate	Kriging
Optimisation	Global search: GA – Random initial population: $150 = 50d$ – Crossover rate: 0.8 – Mutation rate: 0.01 – Replacement rate: 1 – Convergence tolerance: $1.E-4$ – Maximum number of evaluations: 1000
	Local search: PS – Initial points: 3 from GA – Initial step size: $\delta_0 = 0.2$ – Contractor factor: $c = 0.5$ – Convergence tolerance: $\epsilon_{\text{stop}} = 1.E-5$ – Maximum number of function evaluations: 1000 – Maximum number of iterations: 100
Infill	Expected Improvement
	Percentage of initial points over the total – 55% (19 initial points and 16 infill points)
	Convergence criteria – Maximum number of evaluations: 35 – $\boldsymbol{x}$-convergence = 0.05

Table 8.2 shows a summary of all the settings, methods, and algorithms that were selected during the analysis and comparison in Chapter 7 for the SBO.

8.4 Results

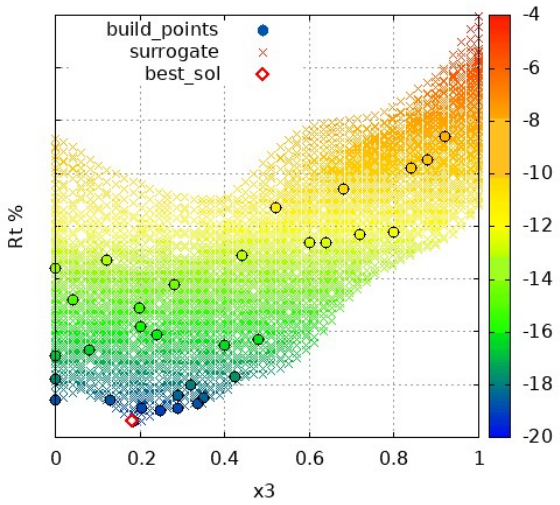
Figure 8.7 represents the summary of results of Surrogate Based Optimisation performed on the Hull Vane structure for the ferry. Each sub-figure represents resistance values, relative to the benchmark, plotted against each normalised variable. In circles, we can see the sampling (both initial and infill) points used to construct the surrogate. The colour of the small x, which are part of the surrogate surface, represents the resistance value for the point in that location. The red diamond is the best (true) value, the output of the routine. The colour

scale represents the resistance variation in reference to the original geometry (where negative values indicate resistance reduction).



(a) *Resistance in kN over x_1 = HV chord length*

(b) *Resistance in kN over x_2 = HV x position*



(c) *Resistance in kN over x_3 = HV z position*

Figure 8.7: Results of Surrogate Based Optimisation performed on the Hull Vane structure for the ferry. In circle the sampling (both initial and infill) points used to construct the surrogate. The colour of the small x, which are part of the surrogate surface, represents the resistance value for the point in that location. The red diamond is the best (true) value, output of the routine.

8.4.1 Design space exploration

The first 19 points of the Halton sequence for a three variables design space are plotted in Figure 8.8. The three histograms demonstrate the space-filling capabilities of the method: the domain is divided into ten equal bands of length equal to 0.1, and in each band for each variable at least one point is present. Furthermore, the distribution is linear, meaning all space has been adequately searched, and no big empty or significant dense areas are present.

The three plots on the lower part of Figure 8.8 represent in two dimensions the three variables design space of the optimisation. For

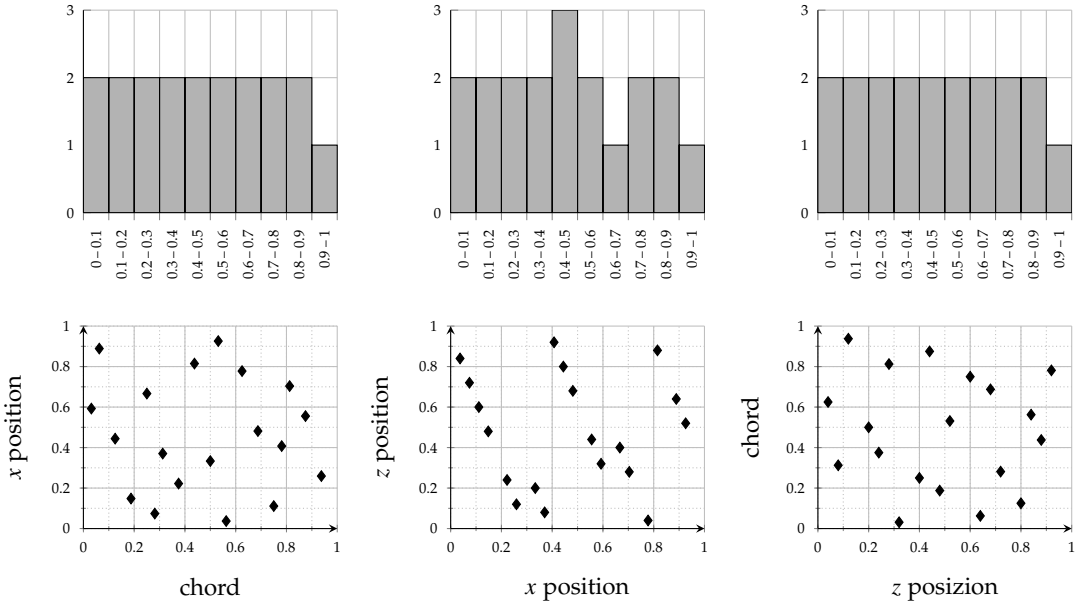


Figure 8.8: First 19 points of the Halton sequence with three variables. Histograms (upper row) show the space-filling capabilities of the sampling plan. Plots (lower row) show the points part of the sequence in the design space used for this optimisation.

each sub-figure one variable is plotted against the other two and the black diamonds are the 19 starting sampling points part of the Halton sequence. As the sequence does not change, these plots are of great importance for the user to check when post-processing. SBO is built on these points and relies its accuracy on the certainty that these points cover the entire domain uniformly. Thus already at first glance, the Naval Architect can check this feature.

A small note must be given for the z position over x position plot. When selecting the Halton sequence, this behaviour that does not look entirely space-filling was known. Moreover, at the time this thesis is written, DAKOTA does not allow a way to bypass this problem. However, if we go back to Figures 7.3, 7.4, 7.5 we can see that for a three variables test function (Hartmann3) such as this case, the Halton sequence was performing better than LHS for each meta-model.

Catching the trend

The first three plots in Figure 8.9 show the resistance results from CFD simulations for the first 19 points of the Halton sequence. Again, the values are given as a percentage reduction in resistance from the benchmark geometry. In each graph, resistance is plotted against one of the three design variables, chord, x position and z position of the Hull Vane. From here, we can already catch the trend of resistance for each variable. For example, one can already see that, for this situation, the best value for chord will be around zero, and the best for z position could fall between 0.2 and 0.4.

This check is crucial to have a first overview of the solution of the optimisation. However, two things are essential to remember here: first, in most cases, the relationship between resistance and one variable

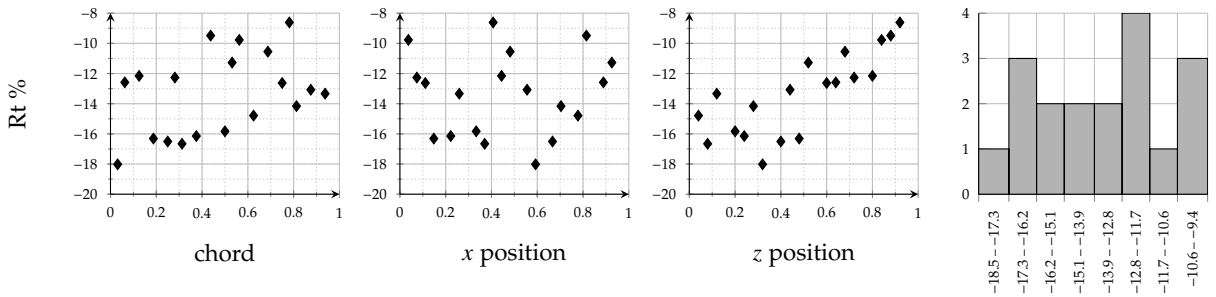


Figure 8.9: Resistance results from CFD simulations of the first 19 points of the Halton sequence plotted against each of the three variables of the optimisation problem. To the right, histogram of resistance distribution for the same points.

is not as easy as linear or quadratic but can be unpredictably non-linear. Second, resistance is influenced by all three parameters and their combinations. Often it is not possible to linearly separate the contribution of each variable. These are precisely the reasons why we do not stop the search for an optimum shape here, but instead, we use a surrogate model to fit these results and predict new possible areas of low resistance. Also, this is why only about half of the total available points are used: the other half will be used by EI to explore/exploit the space in such a way a person would not.

Figure 8.9, on the right, also shows an histogram for resistance values. It can be seen how only one point among the first 19 of the initial sample points had a resistance reduction of over -17.3%.

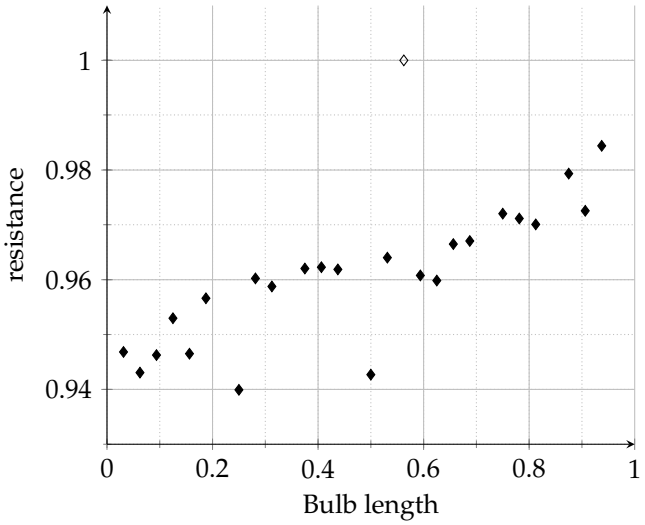


Figure 8.10: Resistance results from CFD simulations of the first 25 points of the Halton sequence plotted against one of the three variables of the optimisation problem: bulb length

Finally, one should not forget about the noise coming from CFD simulations or errors coming from automatic shape modification. From the analysis of these pictures, before going to the next phase of SBO, any anomaly detected should thus be inspected. As an example, we report here resistance results for another optimisation performed at Van Ooossanen N.A.. For this case, the shape of the bulb of a RoRo vessel was optimised considering the length of the bulb, the height of the widest point of the bulb, and the entrance angle of the waterlines at the bulb. Figure 8.10 shows the trend of normalised resistance for the bulb length. It displays how resistance tends to decrease when decreasing the length of the bulb within the design boundaries. However,

there is one point that does not follow the trend (the empty diamond) and has a higher resistance than expected. By analysing the geometry of this point, the Naval Architect would be able to detect an anomaly in the geometry and gain both insights into the problem as well as accuracy for the SBO.

8.4.2 Infill points

The next step is to check the results from the refinement runs. Figure 8.11 shows the position and the resistance values for the refinement points (in red) with respect to the initial points (in black).

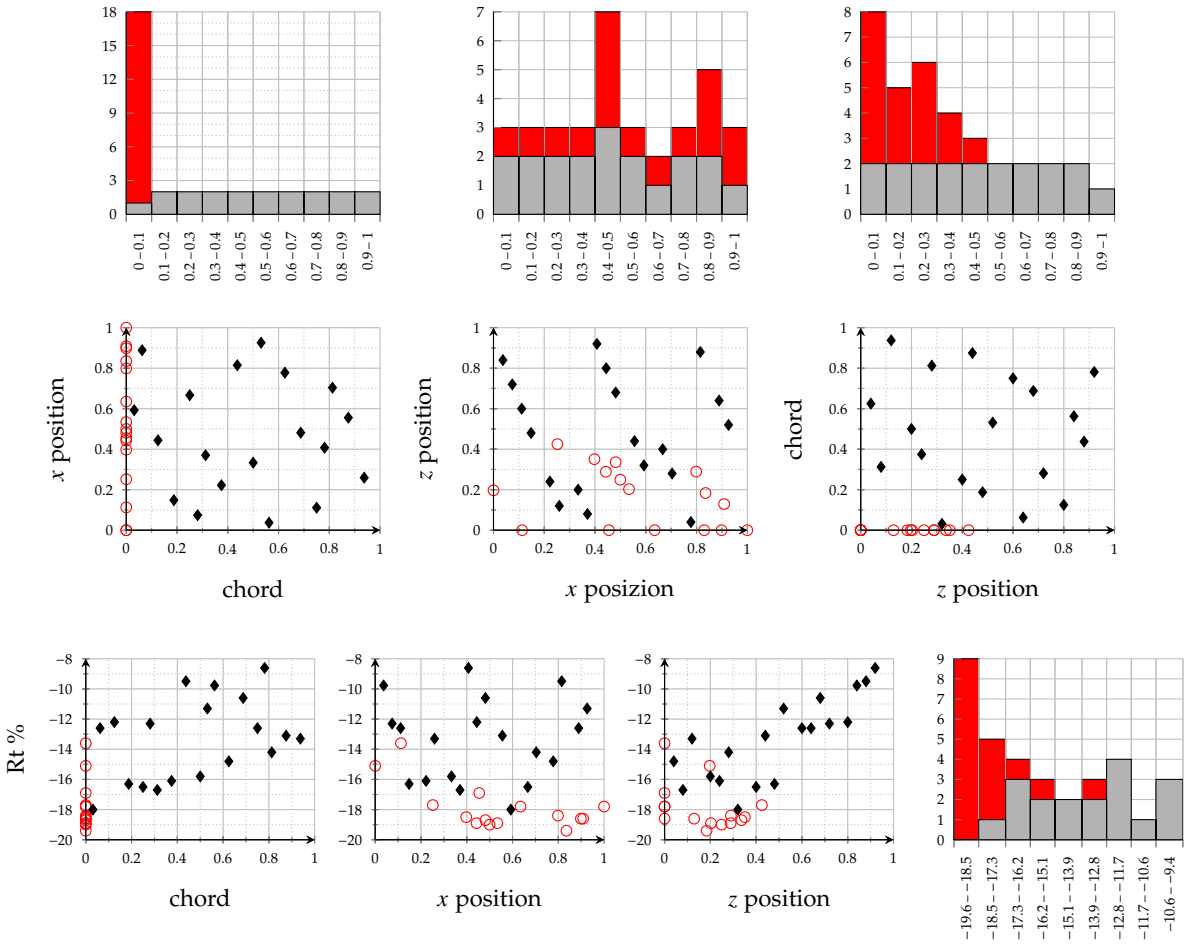


Figure 8.11: Results of refinement runs (in red) with respect to initial results (in black). The upper row represents histograms for HV chord, x position, and z position. The middle row shows the position of all the points in the design space. The lower row represents resistance values plotted against the three design variables and also a histogram for resistance values.

The upper row displays histograms for each of the design variables: HV chord, x position, z position respectively. The Expected Improvement function focused on minimum chord length, as predicted from the trend in figure 8.9. This is because all 16 refinement points are within the 0-0.1 band. Also, it can be seen that for x position of the Hull Vane, the algorithm had to explore the whole domain. While EI could settle and exploit only a smaller area for z position.

The middle row of Figure 8.11 represents in two dimensions the position of the three variables in the design space of the optimisation. In each sub-figure, one variable is plotted against another, and the black diamonds are the same as the one in Figure 8.8, whereas the red circles are the new infill points. The EI function mostly analysed the boundaries of the domain, especially for chord and z position. Nevertheless, it was also able to explore new areas of the graph we were mostly worried about (z over x position), demonstrating how EI can be useful if it is granted with enough points to use for refinement.

The lower row of Figure 8.11 shows the resistance values from CFD simulations with both initial points (black diamonds) and infill points (red circles). In each sub-figure, resistance is plotted over one of the variables, and its value indicates the percentage reduction with respect to the benchmark. It is visible how the EI function was able to "complete" the trend we anticipated in Figure 8.9. Indeed, a local minimum is close to $(0, 0.5, 0.25)$. On the other hand, EI was also able to find a global optimum in an unexpected area. See Table 8.3 for the exact normalised values of this point.

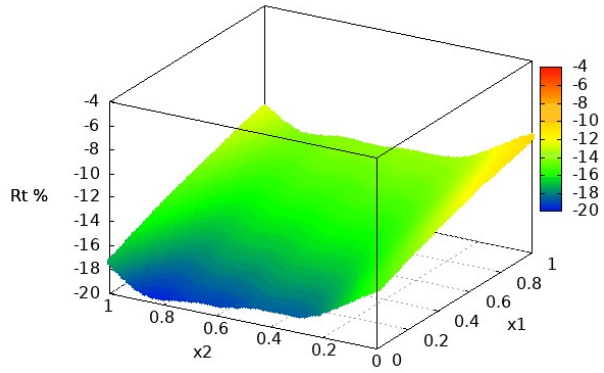
Table 8.3: Best solution for the HV optimisation

	Chord	x position	z position	Rt %
Best Hull Vane	0.000	0.835	0.183	-19.39%

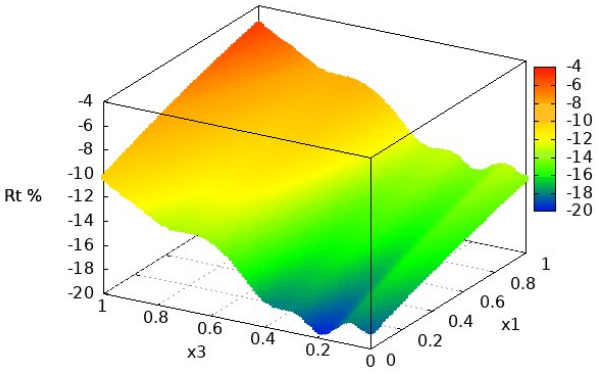
Finally, the histogram on the bottom right corner of Figure 8.11 shows the effectiveness of Expected Improvement function. Only one infill point had a resistance reduction not better than 15.1%; while 13 points out of 16 were found to have equal or lower resistance than the ones in the initial sampling plan.

8.4.3 Surrogate model

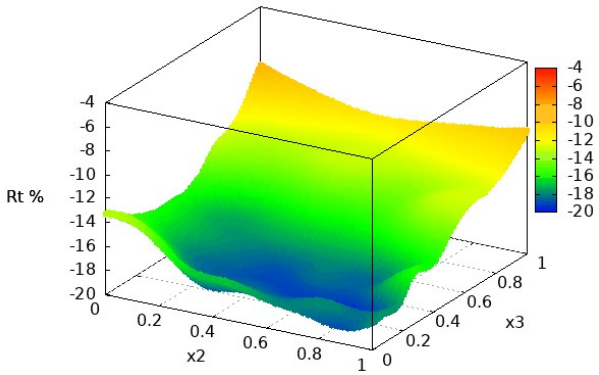
The surrogate model constructed with all the 35 points, both initial and infill ones, is visible in Figure 8.12. Each subfigure represents the percentage resistance reduction with respect to the benchmark plotted over two of the three design variables when keeping the third one fixed to the optimal value found by the optimisation. As an example, for the Figure 8.12a, z position was settled to 0.183. The optimal point is in the blue areas where resistance is the lowest. It can be seen how the approximated function generated by the Kriging model looks smooth, but also able to capture quick changes in curvature when needed. Furthermore, the leave-one-out RMSE error of this prediction with all 35 building points is 1.2%, an acceptably low value.



(a) *Resistance over x_1 = chord and x_2 = x position*



(b) *Resistance over x_1 = chord and x_3 = z position*



(c) *Resistance over x_2 = x position and x_3 = z position*

Figure 8.12: Surrogate model of HV optimisation

8.5 Discussion

The results described in Section 8.4 are now discussed by taking into account two main aspects: the general effectiveness of the SBO routine, and the comparison with the old routine for validation purposes.

8.5.1 Effectiveness of SBO

Table 8.4 shows the percentage of resistance variations found for the geometry with the addition of the optimal Hull Vane with respect to the results obtained with the original geometry.

	Rv	**Rp**	**Rt**
HV opt	+5.80%	-35.65%	-19.39%

Table 8.4: Comparison between resistance values of the Valais benchmark hull and the ones obtained with the best Hull Vane geometry

The optimisation was performed on total resistance (Rt) values, the sum of viscous resistance (Rv) and pressure resistance (Rp). The total resistance does not include corrections due to surface roughness, or aerodynamic resistance, or allowances for sea/service conditions.

From the table, it can be seen that the Surrogate Based Optimisation achieves an overall total resistance reduction of 19.39%. For more clarity of the reader, in the benchmark design the pressure resistance is about the 61% of the total. The differences between the two designs can be better understood by looking at the physics behind those numbers.

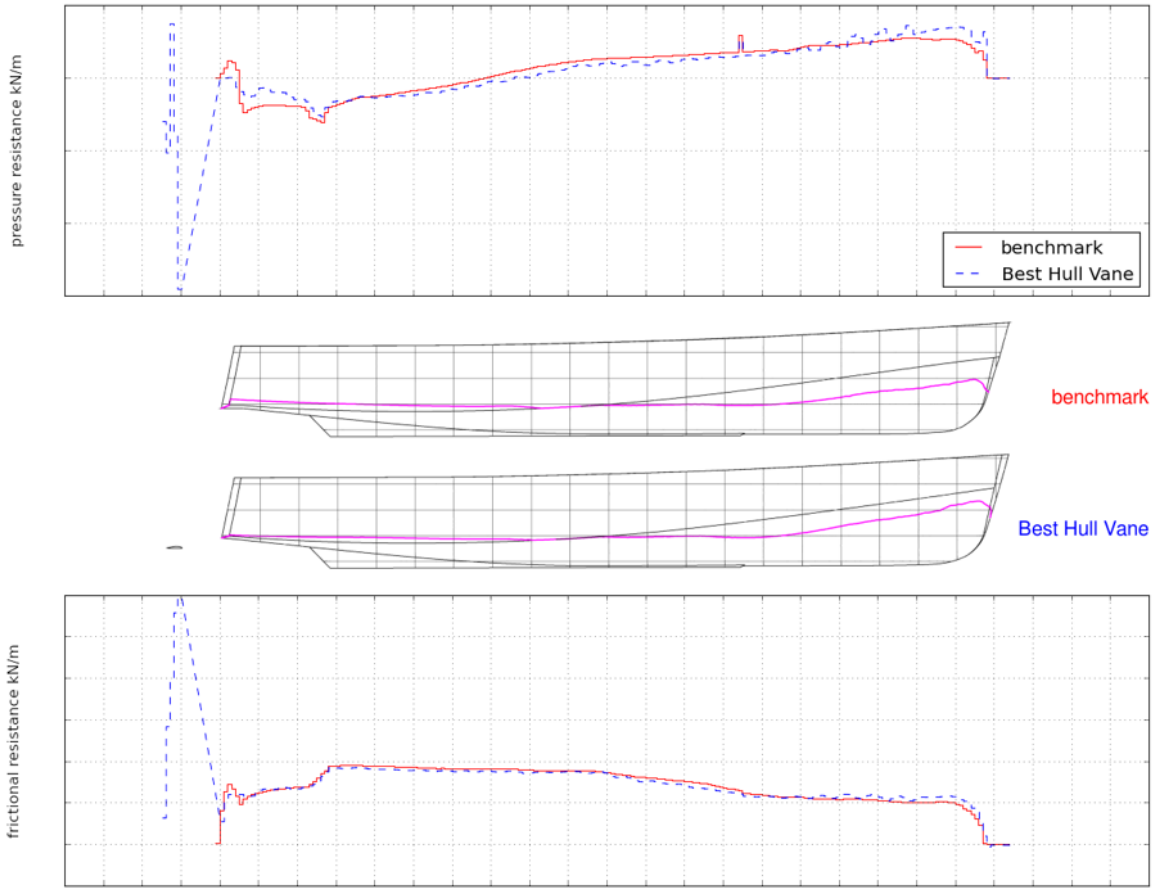


Figure 8.13: Comparison of pressure and frictional resistance for the benchmark vessel and the same geometry with the addition of the best Hull Vane found by SBO. The values are plotted over the length of the vessels, shown in the middle with the pink line representing the free water surface.

First, Table 8.4 shows that there is an increase of 5.8% in viscous resistance when using a Hull Vane. The reason for this is because viscous resistance mostly depends on the wetted surface area when velocity, geometry and fluid properties are not changed. The inclusion of new geometry to the design, the Hull Vane, increases the total wetted surface. In the lowest part of Figure 8.13 it is possible to see the comparison between benchmark (continuous red line) and Hull Vane design (dashed blue line) values of frictional resistance over the length of the vessel. Indeed, a pick of high frictional resistance behind the stern of the hull is present when adding an HV. However, if we compare the viscous resistance values for the hulls only, the difference is only 0.8%, with the benchmark being sightly higher. This difference is because the Hull Vane also slightly modified the trim of the hull.

Second, the most significant reduction is present in pressure resistance, almost -36% less than the benchmark. This loss is mainly due to

the interaction of Hull Vane waves with transom waves. In general, the more water the vessel moves, therefore higher waves are generated, the more energy has to be spent to sail the vessel at the same speed. The result of this interaction is visible in Figure 8.14. The waves behind the vessel in the bottom part of the figure (Hull Vane) are about -58% lower than the ones at the top of the figure (benchmark).

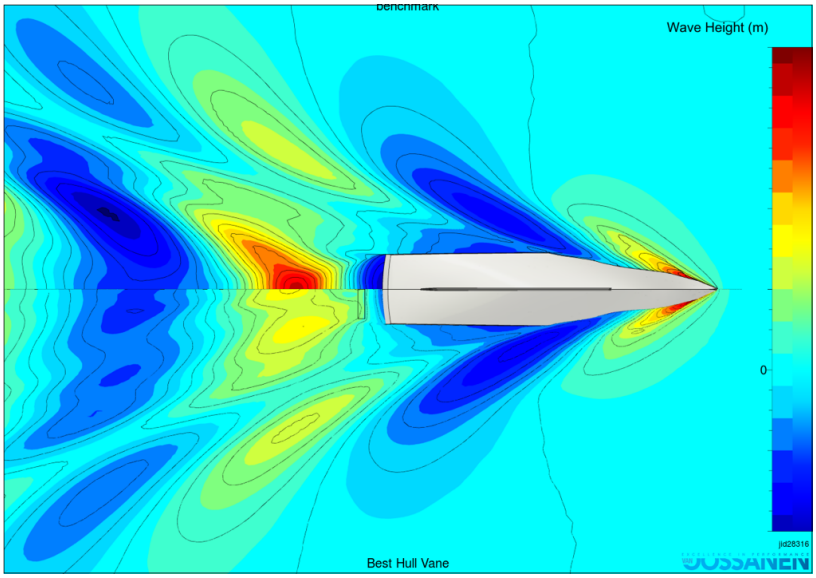


Figure 8.14: Comparison of wave patterns around the benchmark hull and around the new design with same hull geometry and the best HV found by SBO

8.5.2 Validation

Adding a Hull Vane and optimising it with an SBO resulted in a sound resistance reduction for the benchmark geometry. So the Surrogate Based Optimisation already proved to be successful, but mainly thanks to the addition of the Hull Vane itself. Each point calculated during the SBO obtained a better total resistance value than the one from the benchmark. However, to be also validated, the routine has to be compared with the results found by an old optimisation performed with the same simulation settings, and the same geometry. The old results obtained by Van Oossanen N.A. can be used for this validation because the best Hull Vane selected has been installed on the ferry. Speed trials were performed before and after the installation of the Hull Vane and a reduction of 19% of shaft power was experienced when sailing at design speed.

	Rv	**Rp**	**Rt**
Old best results	+7.26%	-33.80%	-19.06%
New best results	+5.80%	-35.65%	-19.39%

Table 8.5: Comparison of total, viscous, and pressure resistance between the benchmark geometry and the best results obtained by the old and the new Surrogate Based Optimisations

Table 8.5 shows the comparison of the total, viscous, and pressure resistance values obtained from CFD for the two SBO routines. The percentages represent a resistance variation in reference to the original geometry (where negative values indicate resistance reduction and

positive values indicate resistance increment). The new routine managed to reduce the total resistance of the benchmark (19.39%) slightly more than the old routine (19.06%). Nevertheless, this improvement is not decisive, because it could fall within the CFD uncertainties.

In Table 8.6 the details of the position and total resistance values obtained with the old and the new SBO routine are shown. We can see that the new routine was able to select the same area around $(x_1, x_2, x_3) = (0, 0.5 - 0.6, 0.25 - 0.3)$ than the old optimisation. Furthermore, the resistance results for those two points varies only of a 0.1%. However, the new SBO was also able to explore the design space better and find another point (the new best point in the table) where total resistance is 0.3% better than before.

Table 8.6: Comparison of position and resistance values found with the old SBO routine and the new one

	Chord	**x position**	**z position**	**Rt %**
Old best point	0.014	0.585	0.297	-19.06%
New best point	0.000	0.835	0.183	-19.39%
Second new best point	0.000	0.499	0.249	-18.97%

Finally, we can say that the routine is validated. It also represents a small improvement, even though at the cost of a higher number of simulations. Moreover, the reason for this is to be found in the fact that the new SBO was able to pass the same point of local minimum where the old optimisation stopped, and moved on to a global optimum with lower resistance. The new routine allowed a better exploration of the design space, thanks to the small increase in total number computations and to the different ratio of initial over total points.

A total of 25-30 points were usually used at Van Oossanen N.A. for a three variables optimisation. Furthermore, the refinement points were only 5, meaning that the ratio between initial and total points is higher than 80%. When in Chapter 7, we demonstrated that the best performance is found with the ratio between 45% and 55%. This can be seen in Figure 8.15, where it is clear that the five refinement points (in blue) exploited the area close to the optimum, but were not enough to explore other areas.

Finally, we should remember that, as shown in Section 2.2, the design of a ship involves a multitude of factors, and each is traded off against each other. Therefore, the choice of which Hull Vane to build should not be based solely on the results of this optimisation. This SBO takes into account only resistance reduction. However, for example, one could also consider that a Hull Vane too far away from the transom (both in x and z direction) is not ideal. Firstly, it is more challenging to construct the structures that attach the Hull Vane to the hull. Secondly, if the HV protrudes too much from the hull, turning, manoeuvring, and docking will have a higher chance of damaging the appendage. The new routine has thus a second advantage compared

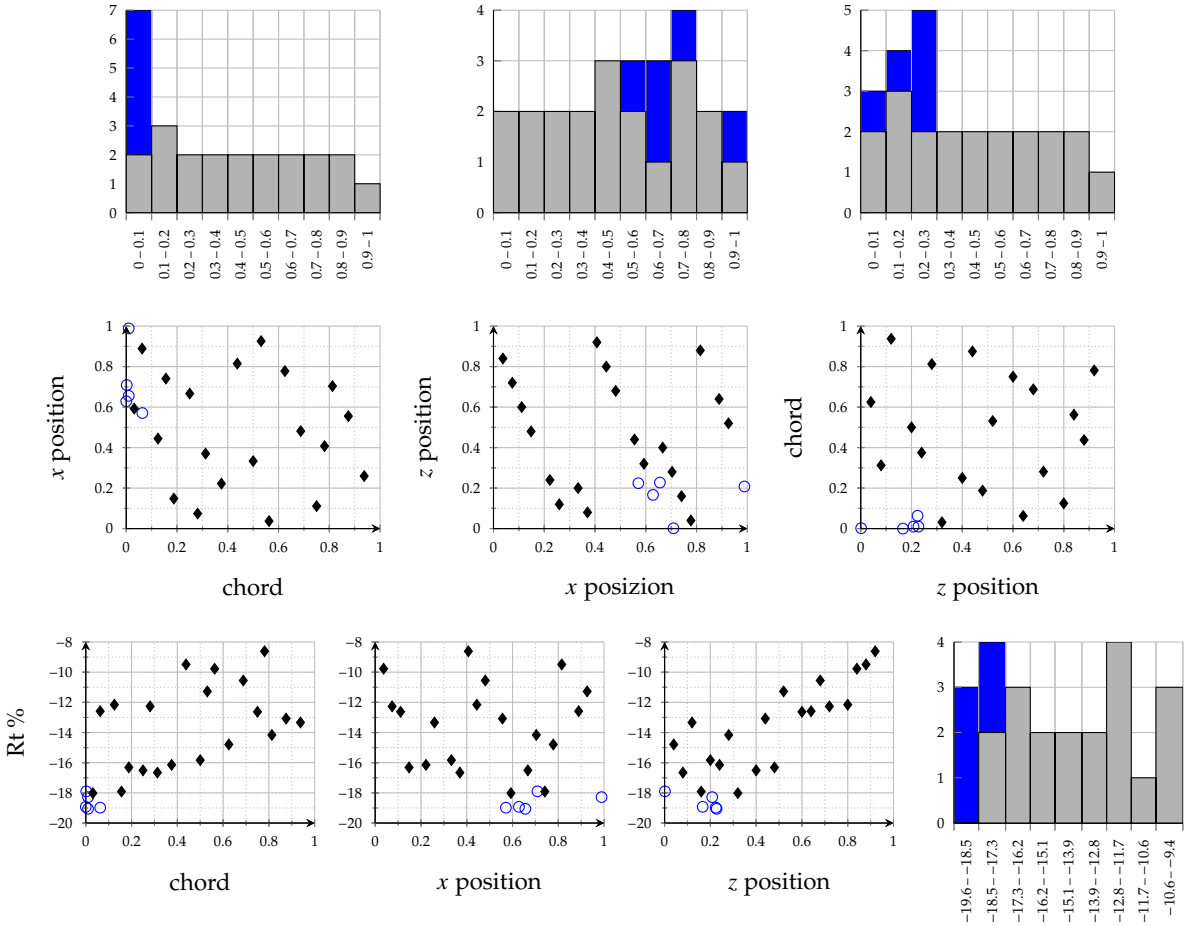


Figure 8.15: Results of refinement runs (in blue) with respect to initial results (in black) for the old SBO. The upper row represents histograms for HV chord, x position, and z position. The middle row shows the position of all the points in the design space. The lower row represents resistance values plotted against the three design variables and also a histogram for resistance values.

to the old one. It shows that there is not just one point, but a whole area where a robust optimal solution can be found (see Figure 8.12c). This way, the Naval Architect has more freedom in considering other objectives rather than just resistance.

Validation test: Aft Ship 9

The second test with real CFD data representing the resistance of a hull shape is an aft ship optimisation case previously studied at Van Oossanen N.A.. In the original case, Van Oossanen N.A. was asked to investigate the aft ship geometry of a inland vessel when sailing in shallow waters. The geometry of the hull is visible in Figure 9.1.

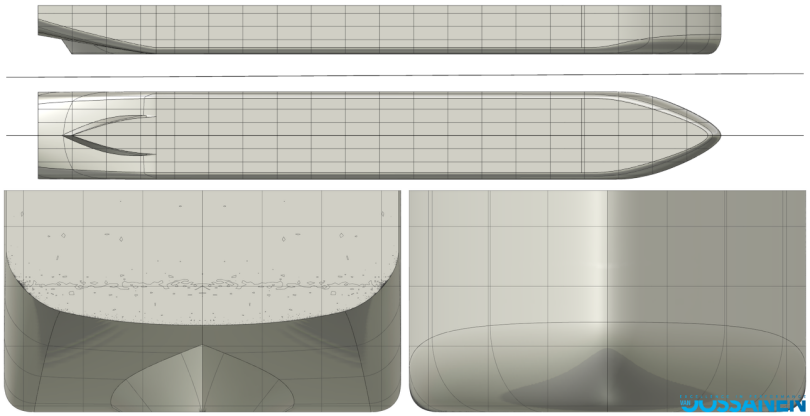


Figure 9.1: Side, bottom, aft and forward view of the geometry

For this test, the same exact simulation settings, the same benchmark geometry, the same variables, and the same modeller for geometry modifications as the original ones are used. All the information regarding the case is a courtesy of the company, and it is part of their official reports. However, due to the intellectual property of Van Oossanen on the data, all the values in this thesis are presented using non-dimensional numbers or percentages.

9.1 Simulation Settings

All numerical simulations described in this thesis are performed with the software tools illustrated in Chapter 1. The purpose of the simulations is to obtain a resistance value for a particular hull geometry. The ship motions that are calculated by 6 degrees of freedom solver were restricted only to allow trim and heave besides the imposed velocity. The simulation is started at zero speed after which the ship is gradually accelerated to the final velocity, using a fourth of a sinusoidal ramp. To achieve a stable solution, 3500 time steps are used.

A "volume of fluid" method is used to account for the free surface (i.e. both water and air flows are solved), for which the parameters are given in Table 9.1. The solver uses the (Unsteady) Reynolds Averaged Navier Stokes (U)RANS equations which describe the flow. These equations need a closure model for which the two-equation $k-\omega$ SST Menter turbulence model was used. The freestream turbulence quantities were initialised using the reference length and velocity. Wall

Table 9.1: Water and air parameters used for the portliner test case

Fluid properties			
Fresh water viscosity	μ_w	0.001217	Pa s
Water density	ρ_w	1000	kg m^3
Air viscosity	μ_a	0.0000185	Pa s
Water density	ρ_a	1.2	kg m^3

functions were used to simulate the flow in regions very close to solid walls, reducing the mesh density requirements in the boundary layer.

9.1.1 Variables and constraints

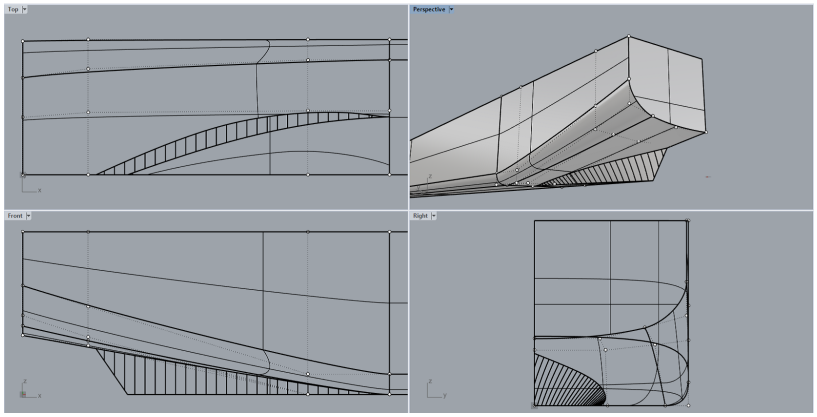


Figure 9.2: Control points in Rhino to modify the aft shape

The coordinate system adopted is the right-handed Cartesian coordinate system, see Figure 8.3. The positive x-direction is defined as the direction from aft to forward (red arrow). The positive y-direction is defined as the direction to the port side of the vessel (blue arrow). The positive z-direction is defined in an upward direction (green arrow). It follows that negative trim is bow-up and positive rise indicates a positive vertical movement of the hull at the centre of gravity. The origin of the coordinate system is located at the baseline at the centre line aft of the transom.

For this optimisation, the aft ship geometry is varied with the use of four parameters. They are: transom submergence, deadrise, shoulder x-position (see Figure 9.3), and S-curve height (see Figure 9.4). In Figure 9.2, the control points on the aft are shown. The geometry is changed by moving the relevant control points. In particular:

Transom submergence: points of the aft side, bilge, and bottom are moved in the z direction by the same amount.

Deadrise: the centre transom point is moved down with a factor times the parameter. The other two control points of the transom are then moved down relative to the start of the bilge, such that the

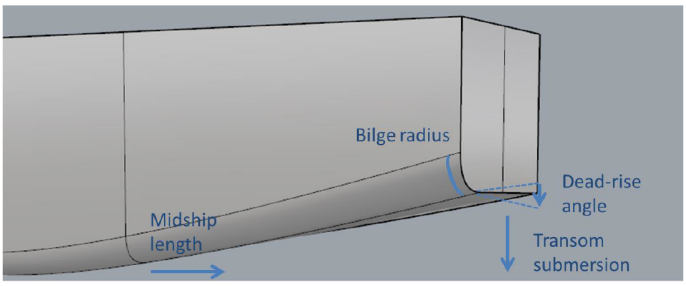


Figure 9.3: Definition of transom submergence, deadrise angle, and midship length

relative z distance between the transom centre and the bilge is constant.

S-Curve height: the last control points of the aft bottom defining the S shape visible in Figure 9.4 are all moved in the z direction of the same amount.

Shoulder X: the two corner points of the bilge radius (Figure 9.2) are moved in y and z direction. Then the points in between are moved accordingly.

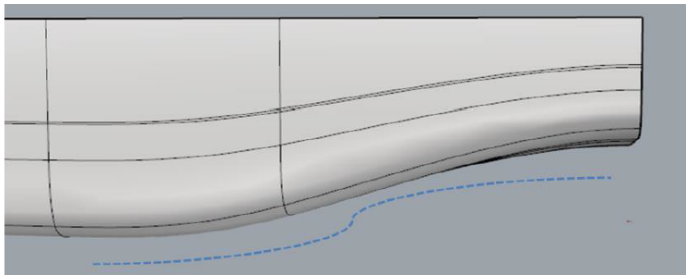


Figure 9.4: Definition of the S aft shape

Some constraints are already integrated into the morphing equations in order to avoid weird or unfeasible shapes. For instance, if the transom submergence is very low, the deadrise angle cannot be too large. The choice of variables, and how they can vary in the space was made in such a way that designs with lousy geometry, impossible shapes or are known beforehand to yield poor performance are avoided. Furthermore, constraints are set up in order not to waste computation time for fruitless candidates.

9.1.2 Domain and mesh

The computational domain is aligned with the reference system and is made up of a rectangular box in which half of the hull geometry was placed. The domain around the hull is constructed such that the boundaries are far enough away so as not to influence the results. In setting up the domain, the default guidelines used by Van Oossanen N.A. were used.

All hull surfaces had a no-slip boundary condition using wall functions to capture the boundary layer. In the symmetry plane a mirror boundary condition was applied, and on the top and the bottom of the domain the pressure was prescribed. All other domain faces have external/free-flow boundary conditions with a prescribed flow speed of $v = 0$ m/s.

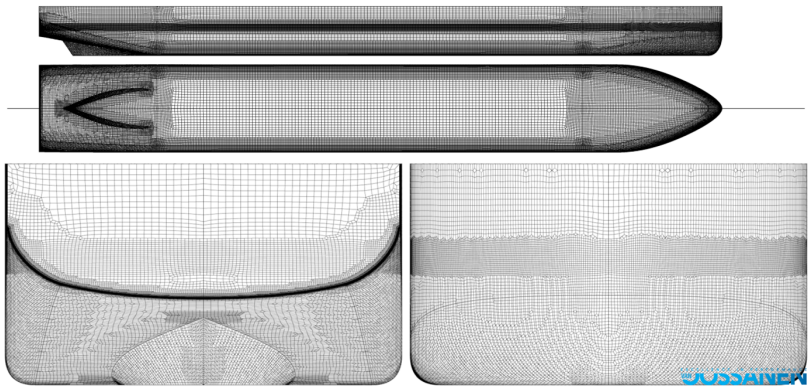


Figure 9.5: Example of mesh computed for the vessel hull

Example of the mesh on the surface of the hull is given in Figure 9.5.

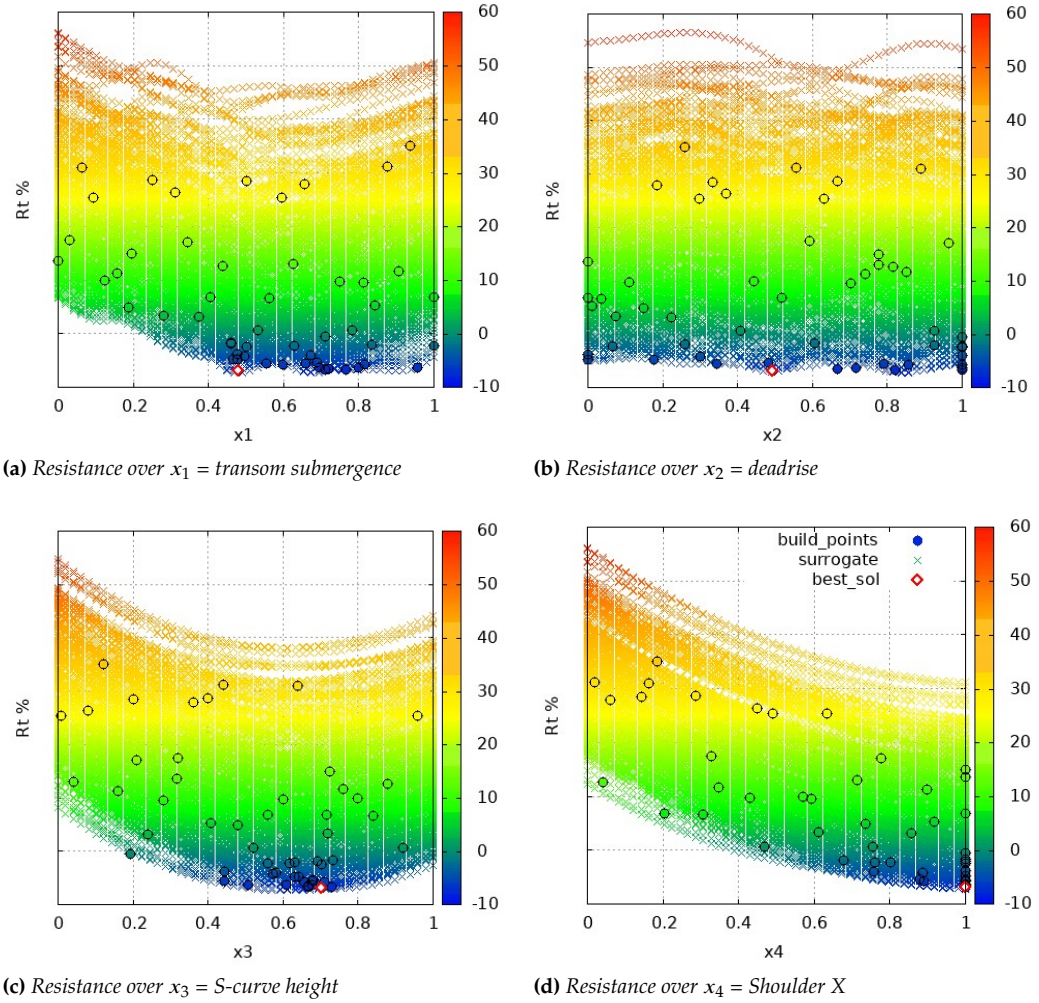
9.2 SBO settings

Table 9.2: Summary of conclusions drawn from Chapter 7 for a $d = 4$ variable optimisation

Item	Adopted solution
Sampling	Initial sample method – Halton sequence
	Total sample size – $60 = 15d$
Surrogate	Kriging
Optimisation	Global search: GA – Random initial population: $200 = 50d$ – Crossover rate: 0.8 – Mutation rate: 0.01 – Replacement rate: 1 – Convergence tolerance: $1.E-4$ – Maximum number of evaluations: 1000
	Local search: PS – Initial points: 3 from GA – Initial step size: $\delta_0 = 0.2$ – Contractor factor: $c = 0.5$ – Convergence tolerance: $\epsilon_{\text{stop}} = 1.E-5$ – Maximum number of function evaluations: 1000 – Maximum number of iterations: 100
Infill	Expected Improvement
	Percentage of initial points over the total – 45% (27 initial points and 33 infill points)
	Convergence criteria – Maximum number of evaluations: 60 – $\boldsymbol{x}$-convergence = 0.05

Table 9.2 shows a summary of all the settings, methods, and algorithms that were selected during the analysis and comparison in Chapter 7.

9.3 Results



(a) *Resistance over x_1 = transom submergence*

(b) *Resistance over x_2 = deadrise*

(c) *Resistance over x_3 = S-curve height*

(d) *Resistance over x_4 = Shoulder X*

Figure 9.6: Results of Surrogate Based Optimisation performed on the Aft geometry of the portliner vessel. In circle the sampling (both initial and infill) points used to construct the surrogate. The colour of the small x, which are part of the surrogate surface, represents the percent difference of resistance with respect to the benchmark for the point in that location. The red diamond is the best (true) value, output of the routine.

Figure 9.6 represents the summary of results of Surrogate Based Optimisation performed on the Aft geometry of the vessel. Each subfigure represents the percentage of difference to the benchmark of total resistance plotted against each normalised variable. In circle shape, the sampling (both initial and infill) points used to construct the surrogate. The colour of the small x, which are part of the surrogate surface, represents the resistance value for the point in that location. The red diamond is the best (exact) value, the output of the routine. The colour scale represents the resistance variation from the original geometry (where negative values indicate resistance reduction and positive values indicate resistance increase).

9.3.1 Design space exploration

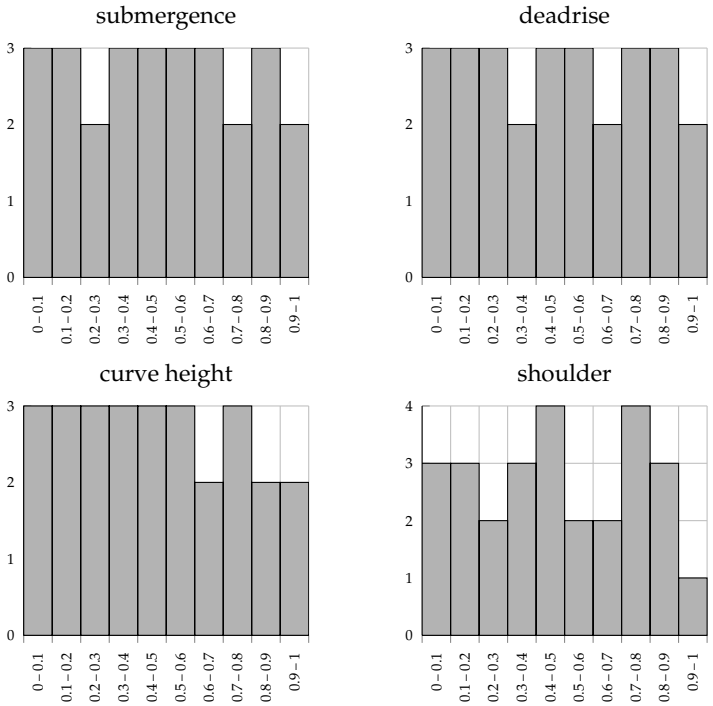


Figure 9.7: First 27 points of the Halton sequence with four variables. Histograms show the space-filling capabilities of the sampling plan.

The first 27 points of the Halton sequence for a four variables design space are plotted in Figure 9.7 and 9.8. First, Figure 9.7 shows four histograms that demonstrate the space-filling capabilities of the method: the domain is divided into ten equal bands of length equal to 0.1. With only one exception, in each band for each variable, at least two points are present. Furthermore, the distribution is mostly linear, meaning all space has been adequately searched, and no too empty or too dense areas are present.

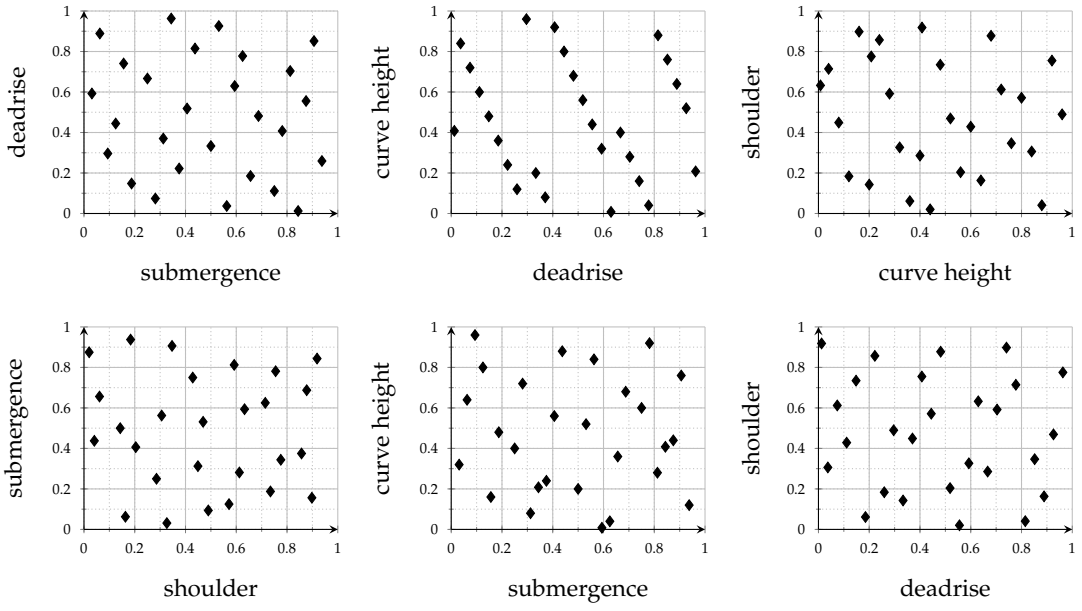


Figure 9.8: First 27 points of the Halton sequence with four variables. Plots show the points part of the sequence in the design space used for this optimisation.

Second, the six plots on Figure 9.8 represent in two dimensions the four variables design space of the optimisation. For each sub-figure one variable is plotted against another and the black diamonds are the 27 starting sampling points. As the sequence does not change, these plots are of great importance for the user to check when post-processing. SBO is built on these points and relies its accuracy on the certainty that these points cover the entire domain uniformly. Thus already at first glance, the Naval Architect can check this feature.

A small note must be given for the curve height over deadrise plot. When selecting the Halton sequence, this behaviour that does not look entirely space-filling was known. It was also already noticed in Figure 8.8. However, the analysis of Chapter 7 and the validation test in Chapter 8 already proved that this is not necessarily an issue.

Catching the trend

The four plots in Figure 9.9 show the resistance results from CFD simulations for the first 27 points of the Halton sequence. In each graph, resistance is plotted against one of the four design variables, transom submergence, deadrise, S-curve height, shoulder X. From here, we can already catch the trend of resistance for each variable. For example, one can already see that, for this situation, the best value for shoulder will be around one, and the best for curve height could fall between 0.6 and 0.8.

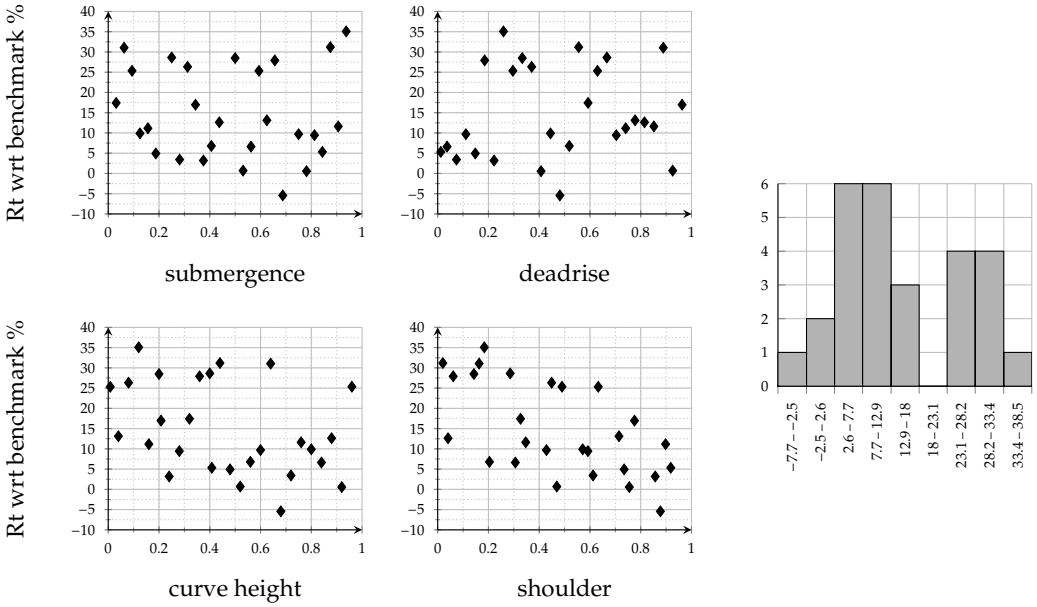


Figure 9.9: Resistance results from CFD simulations of the first 27 points of the Halton sequence plotted against each of the four variables of the optimisation problem. To the right, histogram of resistance distribution for the same points.

This check is vital to have a first overview of the solution of the optimisation. However, two things are essential to remember here: first, in most cases, the relationship between resistance and one variable is not as easy as linear or quadratic but can be unpredictably non-linear. Second, resistance is influenced by all four parameters and

their combinations. Often it is not possible to linearly separate the contribution of each variable. These are precisely the reasons why we do not stop the search for an optimum shape here, but instead, we use a surrogate model to fit these results and predict new possible areas of low resistance. Also, this is why only about half of the total available points are used: the other half will be used by EI to explore/exploit the space in such a way a person would not.

Figure 9.9, on the right, also shows an histogram for resistance values. It can be seen how only one point among the first 27 of the initial sample points had a resistance value lower than the benchmark (0%). Finally, one should not forget about the noise coming from CFD simulations or errors coming from automatic shape modification. However, from the analysis of these pictures, no evident anomaly is detected.

9.3.2 Infill points

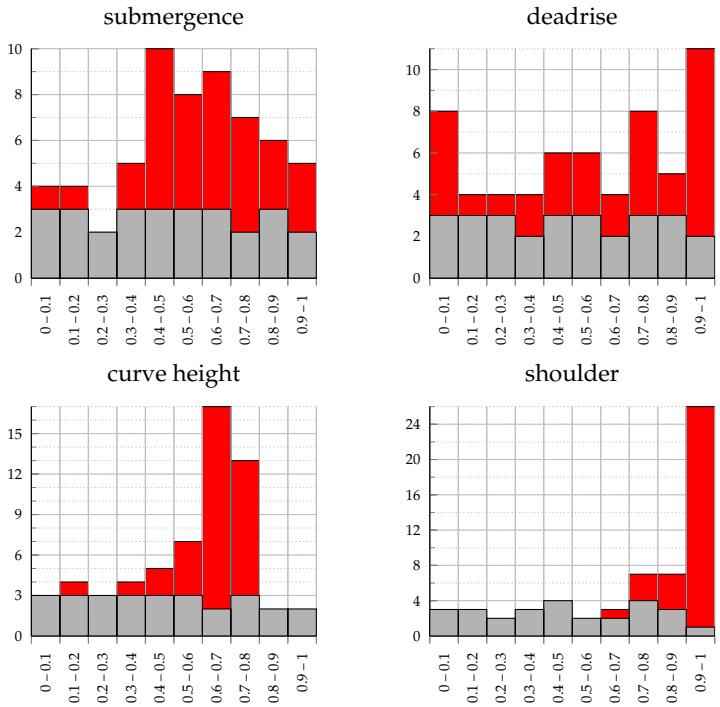


Figure 9.10: Results of refinement runs (in red) with respect to initial results (in black) in the form of histograms for submergence, deadrise, curve height, shoulder variables.

The next step is to check the results from the refinement runs. Figures 9.10 displays histograms for each of the design variables: transom submergence, deadrise, S-curve height, shoulder X. The Expected Improvement function focused on the maximum shoulder. As predicted from the trend in Figure 9.9. This is because 25 refinement points out of 33 are within the 0.9-1 band. Also, as anticipated in Figure 9.9, EI could settle and exploit only a smaller area for curve height, between 0.6 and 0.8. On the other hand, for submergence and deadrise, the algorithm had to explore the whole domain.

Figure 9.11 represents in two dimensions the position of the points in the four variables design space of the optimisation. In each subfigure, one variable is plotted against another, and the black diamonds are the same as the one in Figure 9.9, whereas the red circles are the

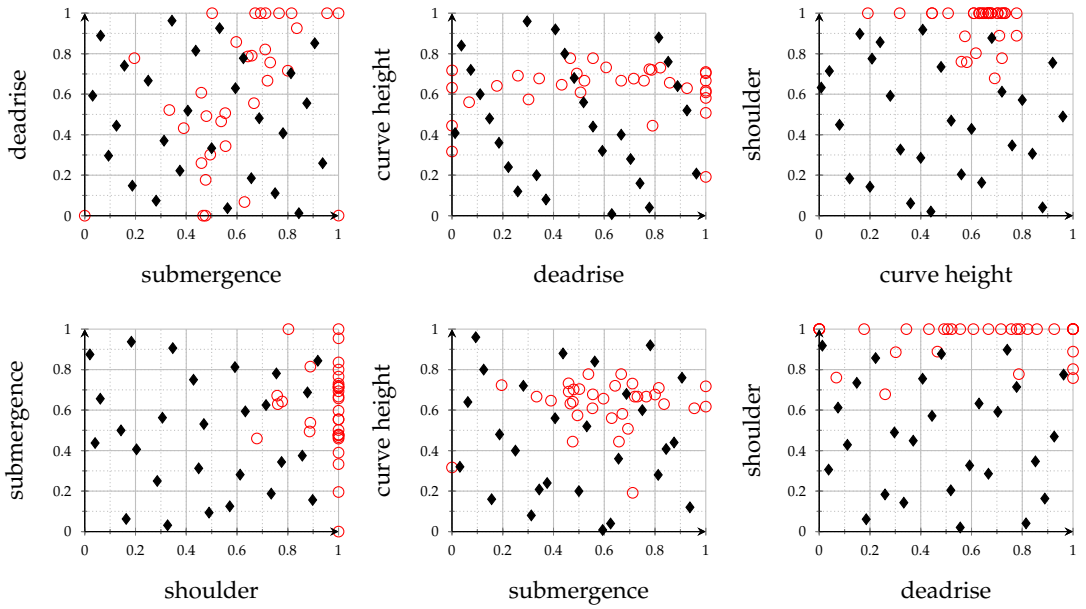


Figure 9.11: Results of the position of all the refinement points (in red) in the design space with respect to initial results (in black).

new infill points. The EI function mostly analysed the boundaries of the domain for the shoulder variable. Nevertheless, it also looks like that it struggled in figure out a single area where deadrise is best, especially when coupled with submergence.

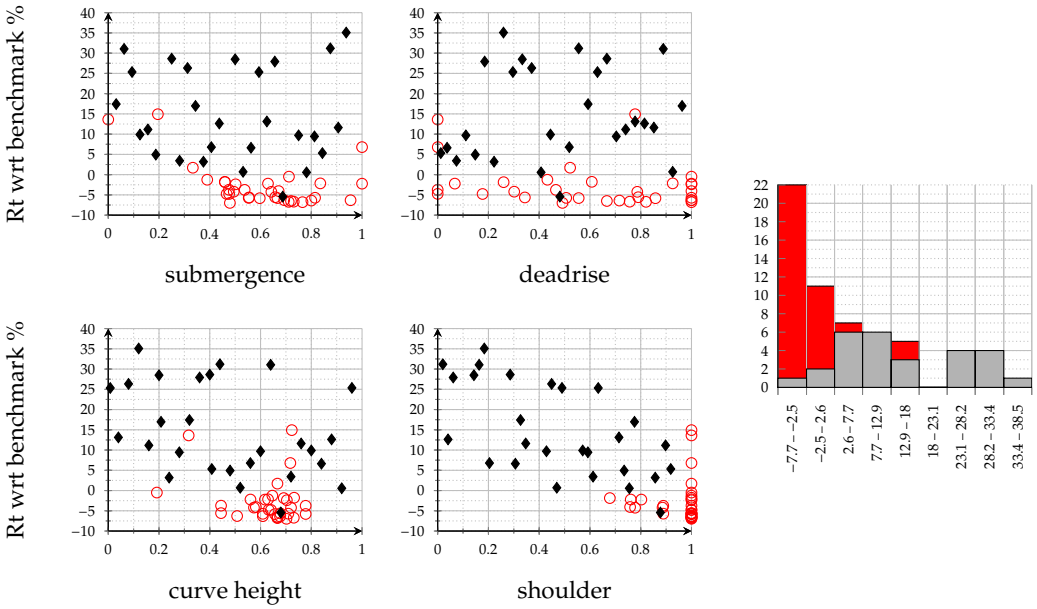


Figure 9.12: Results of refinement runs (in red) with respect to initial results (in black). To the left resistance values are plotted against the four design variables and to the right in displayed a histogram for resistance values.

Figure 9.12 shows the resistance values from CFD simulations with both initial points (black diamonds) and infill points (red circles). In each sub-figure, resistance is plotted over one of the variables. It is visible how the EI function was able to "complete" the trend we anticipated in Figure 9.9 for three out of four variables. The most problematic one is deadrise, that does not provide a clear trend for

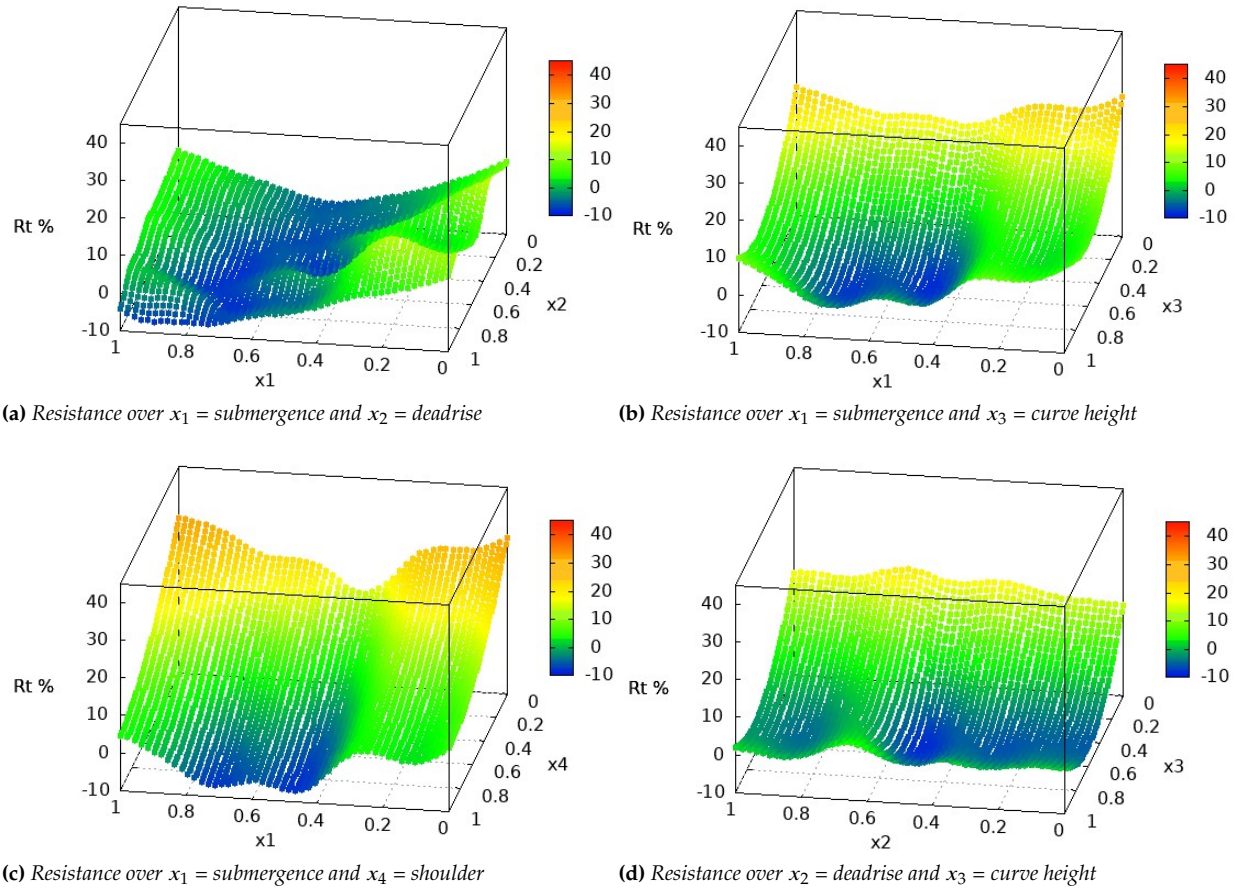
resistance. A local minimum is close to (0.75, 1.0, 0.7, 1.0). However, EI was also able to find a global optimum in a different area. See Table 9.3 for the exact normalised values of this point.

Table 9.3: Best solution for the AFT optimisation

	submergence	deadrise	curve height	shoulder	Rt %
Best solution	0.479	0.492	0.702	1.000	-6.98%

Finally, the histogram on the right side of Figure 9.12 shows the effectiveness of Expected Improvement function. Only three infill point had an increase in resistance with respect to the benchmark (more than 0%). Moreover, 21 points out of 33 were found to have equal or lower resistance than the ones in the initial sampling plan.

9.3.3 Surrogate model



(a) *Resistance over x_1 = submergence and x_2 = deadrise*

(b) *Resistance over x_1 = submergence and x_3 = curve height*

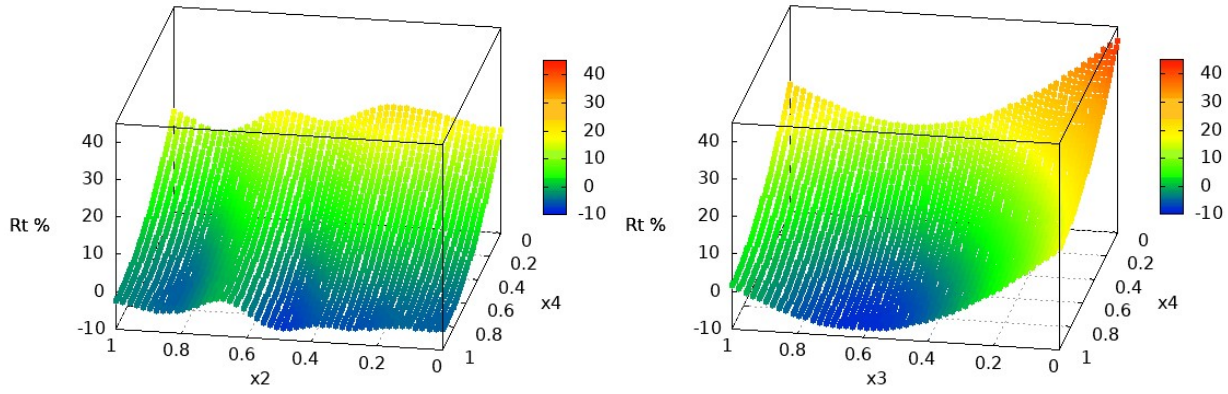
(c) *Resistance over x_1 = submergence and x_4 = shoulder*

(d) *Resistance over x_2 = deadrise and x_3 = curve height*

Figure 9.13: Surrogate model of AFT optimisation

The surrogate model constructed with all the 60 points, both initial and infill is visible in Figure 9.13. Each subfigure represents the resistance plotted over two of the four design variables, when keeping

the other two fixed to the optimal value found by the optimisation. As an example, for Figure 9.13a, submergence and deadrise are free to change, while curve height was settled to 0.702 and shoulder was fixed to 1.000. The optimal point is in the blue areas where resistance is the lowest. It can be seen how the approximated function generated by the Kriging model looks smooth, but also able to capture quick changes in curvature when needed. Furthermore, the leave-one-out RMSE error of this prediction with all 60 building points is 1.7%, an acceptably low value.



(e) *Resistance over x_2 = deadrise and x_4 = shoulder*

(f) *Resistance over x_3 = curve height and x_4 = shoulder*

Figure 9.13a: (continued) Surrogate model of AFT optimisation

9.4 Discussion

The results described in Section 9.3 are now discussed by taking into account two main aspects: the general effectiveness of the SBO routine, and the comparison with the old routine for demonstrating it is an improvement.

9.4.1 Effectiveness of SBO

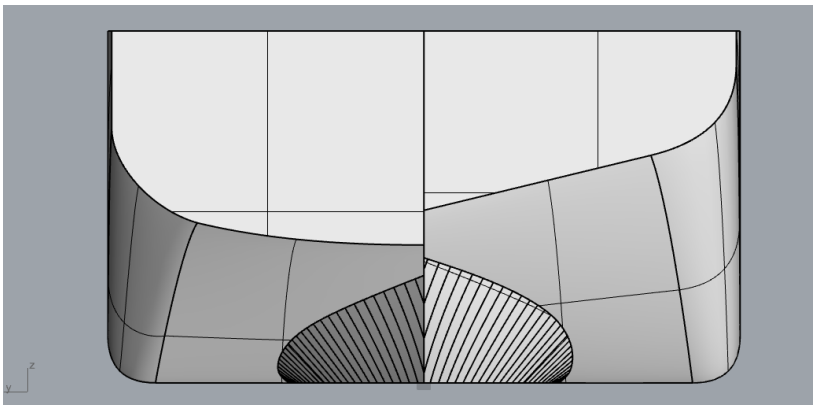


Figure 9.14: Comparison between benchmark and optimised aft geometries

Figure 9.14 shows on the left the benchmark aft geometry and on the right the one found by the Surrogate Based Optimisation. The two

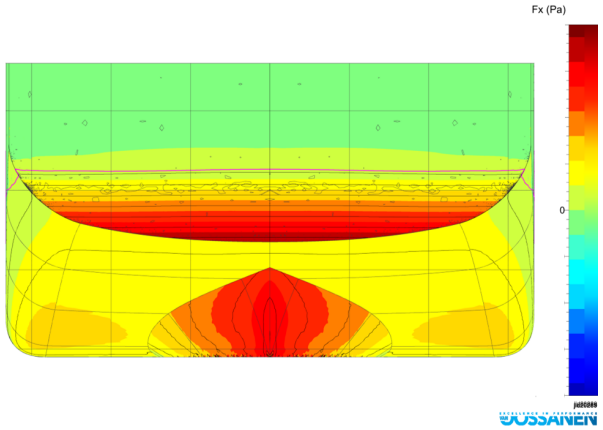
most distinguishable differences are the higher S-curve height and the lower central transom point with respect to the bilge of the new design compared to the benchmark one.

Table 9.4: Comparison between resistance values of the portliner vessel benchmark hull and the ones obtained with the best Aft geometry

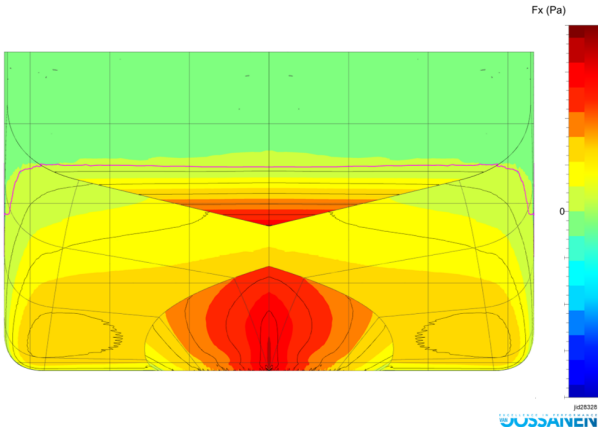
	Rv	Rp	Rt
AFT opt	-0.68%	-11.24%	-6.98%

Table 9.4 shows the values of resistance found both for the benchmark hull and for the optimised geometry with the aft parameters given by SBO. The optimisation was performed on total resistance (Rt) values, the sum of viscous resistance (Rv) and pressure resistance (Rp). The total resistance does not include corrections due to surface roughness, or aerodynamic resistance, or allowances for sea/service conditions.

From the table, it can be seen that the Surrogate Based Optimisation achieves an overall total resistance reduction of 6.98%. For more clarity of the reader, in the benchmark design the pressure resistance is about the 60% of the total. The differences between the two designs can be better understood by looking at the physics behind those numbers.



(a) *Benchmark*



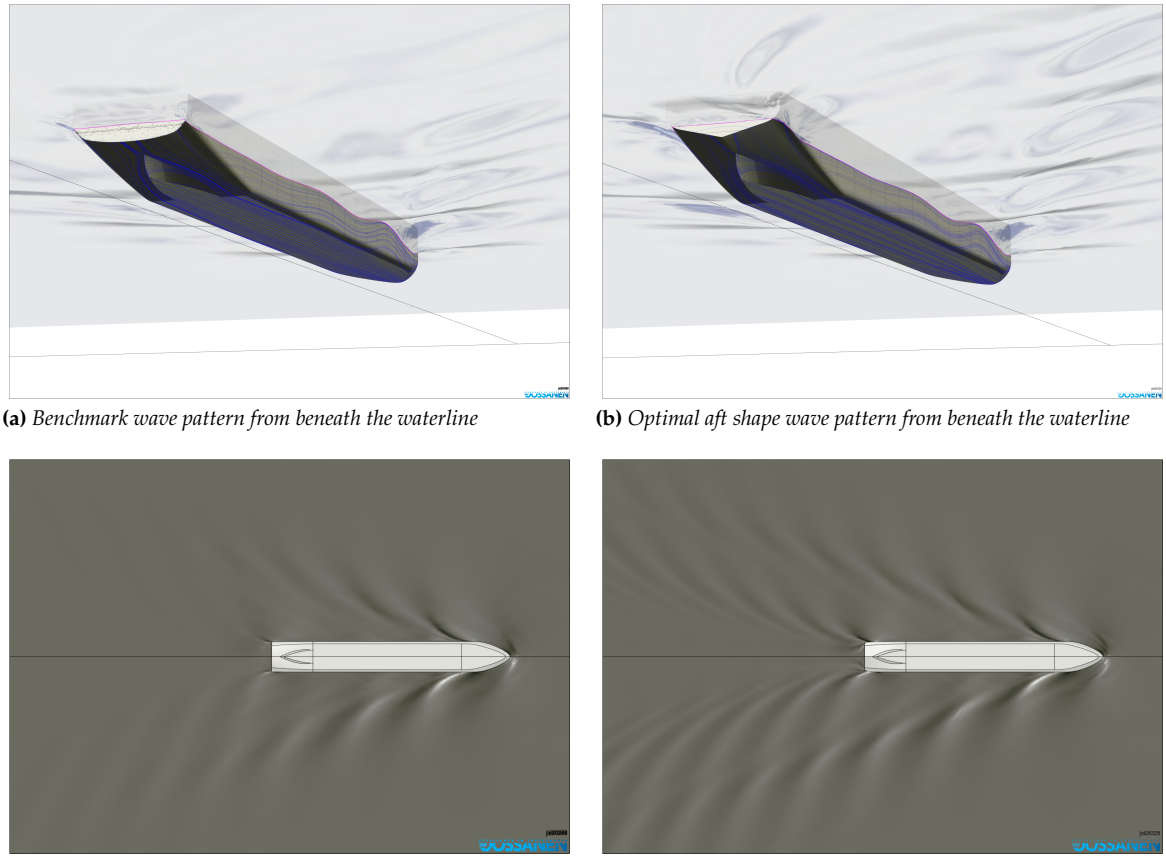
(b) *Optimised AFT*

Figure 9.15: Comparison of Fx contours measured in Pa for the benchmark and the optimised aft geometries.

First, Table 9.4 shows that there is a small reduction of 0.68% in viscous resistance. The reason for this is because viscous resistance

mostly depends on the wetted surface area when velocity, length and fluid properties are not changed. The new geometry has a smaller wetted surface than the benchmark, -0.98%. The difference is mostly due to the different aft shapes, as the trim is almost the same.

Second, the most significant reduction is present in pressure resistance, -11.24% with respect to the benchmark. This could be explained by the big difference between the two transom wetted areas (see Figure 9.15). As stated by Holtrop and Mannen in 1982 [37], the total resistance of a vessel can be subdivided into several components, of which one is additional pressure resistance due to immersed transom stern. In particular, for this case, the optimised aft managed to lower down the immersed transom area of -51.15%. This is only an approximation, and mostly useful for statistical and regression analysis, but still gives the idea of where the most significant part of resistance reduction comes from.



(a) *Benchmark wave pattern from beneath the waterline*

(b) *Optimal aft shape wave pattern from beneath the waterline*

(c) *Benchmark wave pattern from above*

(d) *Optimal aft shape wave pattern from above*

Figure 9.16: Comparison of benchmark and best AFT shape wave patterns

Finally, when post-processing and analysing CFD results for the optimal aft shape found by SBO, it became clear that the resistance reduction comes with the price of a worse wave pattern than the benchmark. It can be seen in Figure 9.16, where to the left there is the benchmark hull, and to the right the optimised one. The new aft shape generates a second wave pattern at the transom edge, resulting in higher waves and energy loss. This issue happened because the geometry is automatically generated, and it is bound to the morphing

equations. Thus, the recommendation is either to modify the modeller and improve the quality of altered geometries or to correct the optimised solution manually.

9.4.2 Displacement constraint

This case demonstrated that applying certain modifications to the aft shape of a benchmark hull and analysing the results with a Surrogate Based Optimisation can lead to resistance reductions. Thus, once more, the routine build with the analysis carried out in Chapter 7 is successful and validated. However, no constraints regarding displacement were taken into account.

In the original case, the starting points were selected with Halton sampling plan as well, but they did not follow the sequence. Many geometries coming from points of the sequence were discarded because of the displacement constraint. The value had to stay within a small bandwidth. The consequence of this choice is that the sampling plan is not entirely space-filling within the set boundaries. See, for this, Figure 9.17 that shows the starting points in black diamonds and the refinement points in blue circles of the old optimisation plotted for all four variables. Sometimes, it seems clear which areas were neglected by the script, like the upper right angle of shoulder versus curve height plot. Sometimes it is less evident, like in the shoulder over deadrise plot. Consequently, it is not possible to use the entire design space, but it is also not possible to recognise upfront which points will be discarded and which not. Furthermore, this solution is not allowing the Kriging model and the Expected Improvement function to know about the additional constraint.

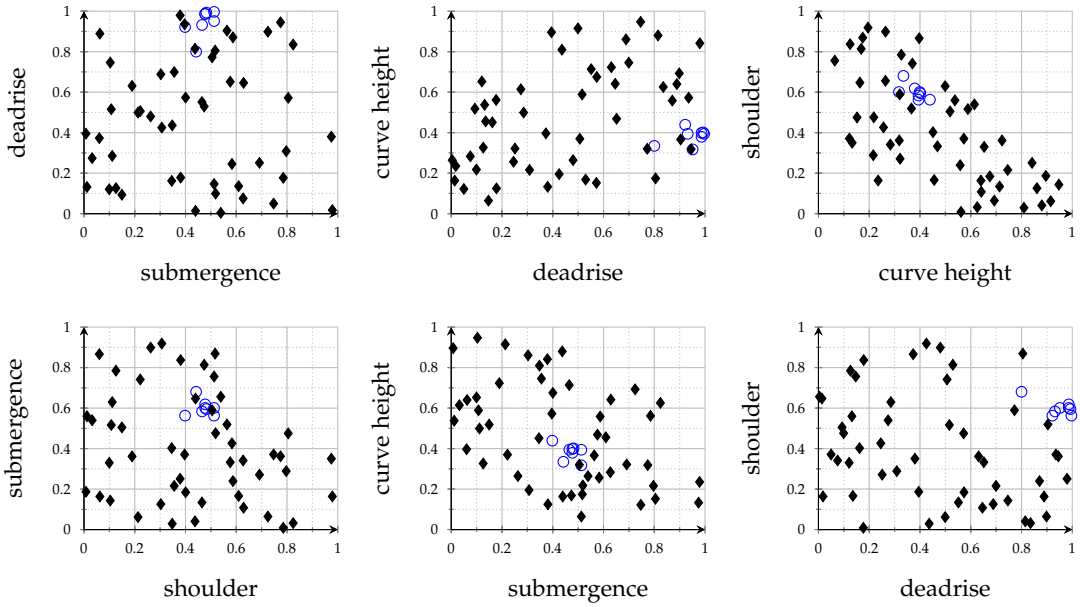


Figure 9.17: Results of the position of all the refinement points (in blue) in the design space with respect to initial results (in black) for the old optimisation routine.

The old optimisations routine did not manage to find a lower resistance for the aft ship. The company pointed out several possible

reasons for this. In particular, they focused on the strong non-linear connection between the variables and in the optimisation routine in general. In their opinion, this can be resolved only with a significant number of other runs that were never performed because of budget limitations. However, we have now more insight into the problem, and we can still prove two points from here.

Firstly, this test allows us also to validate the choice of the Halton sequence and proves that a careful choice of variables and boundaries is crucial. This is because, if the entire domain could be used, then all of it should be sampled. Secondly, the number of total CFD simulation is almost the same (58 for the old and 60 for the new routine), but the ratio between initial and refinement points is entirely different. This thesis proves that about 45%-55% initial points over the total available generate a balanced and successful search of the global optimum. On the other hand, the old routine was starting with over 86% of the total number of points. Therefore, the Expected Improvement function could work with only 8 points, focus on one small area for each variable and not explore any other.

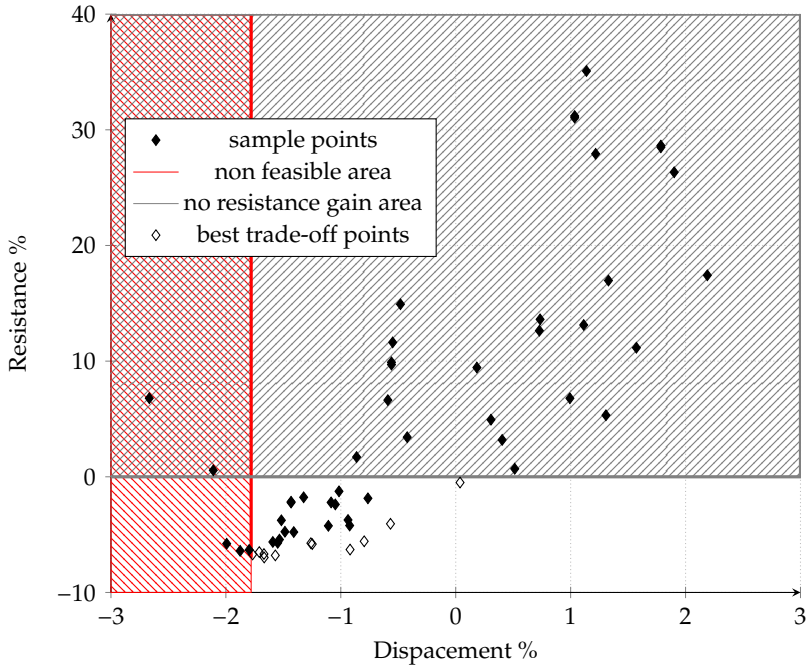


Figure 9.18: Two-objective (resistance and displacement) analysis

Nonetheless, we acknowledge the need for a client to require a displacement constraint, but we recommend to solve the issue in a different way. Either the constraint should be added in the morphing equations directly, thus changing the variables and generating only acceptable designs with a fixed required value. This way, we do not force the optimiser to discard points and lose the space-filling property. Alternatively, another solution could be to have the same four parameters but to use a fifth one (for example, draft) to adjust the draft before sending the geometry to the simulation phase. Alternatively, even, one can consider displacement as another objective to optimise, to maximise in order to allow more cargo. In this last case, the best solution would not necessarily be the one with the lowest resistance,

but the geometry with the best trade-off between two objectives. Figure 9.18 shows the relationship of resistance over displacement for the vessel analysed in this chapter. In diamond shape, both filled and empty, we can see the 60 total sample points. The red area is considered non-feasible as displacement would be too low. The grey area is where the geometry has higher resistance than the benchmark, thus non-acceptable. Hence, the chosen geometry could be one of the empty diamonds as they have lower resistance than the other points with the same displacement. Which one is the best overall is not unique and always valid; it depends on which variable between minimum resistance and maximum displacement weights more in their trade-off analysis.

In conclusion, also this second validation test was successful as a new geometry with less resistance than the benchmark was found. The new ratio of initial versus total available points proved to lead to a proper exploration-exploitation balance. The Halton sequence and its space-filling property are a right choice as, at a glance, the user can already check some results. Furthermore, we proved once more that the post-processing visualisation tools are of great importance. Because, in the end, it does not matter how automatic the optimisation is if the Naval Architect cannot correctly check the results, gain knowledge from them, and spot anomalies. Finally, this test is proof that the design of a vessel is complicated, and the search for lower resistance is essential but bound to many other objectives.

Final remarks

Conclusion 10

This document described the work conducted for the Ship Hydromechanics Master Thesis performed at Van Oossanen Naval Architects company for TU Delft University. The project proposes a methodology for a Surrogate Based Optimisation (SBO) study and its application in the Naval Architecture field using the open-source software DAKOTA. Surrogate Based Optimisation is a useful and powerful way of reducing time and costs in hull shape optimisation. It provides a better global understanding of the problem than traditional optimisation methods. Moreover, it allows the use of exploratory data analysis techniques and machine learning methods to get more insight, discover hidden patterns, and detect anomalies in the design space. However, this procedure has no certainty of convergence, and the result may be not reducing the overall hull resistance.

During the literature review phase, we saw that many authors proved an SBO to be successful and useful, both in the marine field and with the use of DAKOTA. However, no standard rules were ever proposed for how to proceed. Therefore the thesis showed guidelines to follow when building up an SBO routine for a specific topic (resistance reduction by hull geometry modification) and a specific company (Van Oossanen N.A.).

SBO consists of several interconnected steps, and each increase complexity and uncertainty to the overall process. The steps were divided into three sub-topics and further analysed. They are optimisation algorithms, design space exploration (both sampling plans and infill techniques), and surrogate models. The central research part is devoted to the study and comparison of possible methods, their advantages and drawbacks, and their performance, for every sub-topic identified.

First, an overview of what is optimisation is given. The main issue was to select a suitable algorithm, but the number of candidates is high. The most balanced solution for this case is a combination of a global search algorithm that selects promising points and passes them to a local search algorithm that only looks in the vicinity of those. For the global search, Genetic Algorithm is selected, because it is widely used in the literature, it can be adapted to a Multi-Objective optimisation, and it can pass multiple solutions to the next algorithm. As for the local search, since the optimisation is performed on the (continuous) surrogate level, it does not matter which method is chosen. Thus the decision of which one to use was not based on performance, but visualisation capabilities. Pattern Search algorithms, from this point of view, are easy to inspect by the user, thanks to its fixed structure.

Second, the design space exploration is analysed and split into two sub-categories: sampling plans and refinement runs. The two

main problems with sampling plans are the size of the sample nature. Nature is represented by the distribution of points over the design space, and the best way to do it is by using a space-filling technique. Among others, the Halton sequence is preferred as it has no random components. For this reason, t it is easier to spot and correct possible mistakes. Furthermore, it was already used at Van Oossanen N.A.; thus, the company members are already familiar with it. About the second issue, i.e. how many points to use, the higher the number of real points constructs a surrogate, the higher the overall quality will be. Nevertheless, this number is profoundly limited by the budget of the company, because each point requires a long and expensive CFD simulation. A right balance was found with the use of $11d$-$15d$ total points, with d being the number of variables.

Then, the infill sampling criteria (how to improve the accuracy of the surrogate adding points to the initial sampling plan) is described. The biggest concern is how to obtain the right balance between exploration and exploitation of the design space. This balance is achieved with the use of the Expected Improvement function. This algorithm selects the position of a new point based on the uncertainty of the surrogate prediction. It has proved to be able to converge to a global optimum but also to escape a local one if enough iterations are available. Thus, the following question is what percentage of the total points should be used for this refinement phase. The answer is between 45% and 55%.

Third, a generic discussion and overview of surrogate models are given. Many different models are available, but in the literature mostly Kriging, Radial Basis Functions, and Artificial Neural Networks stood out. Among them, Kriging was chosen thanks to its performance measured by a leave-one-out RSME analysis. More than that, KG is also robust, it can deal with non-linearity, and it well supported in DAKOTA toolkit. Two more reasons lead to chose KG over the others. Firstly, this model supports the use of EI functions. Secondly, the expected noise level in Van Oossanen N.A. CFD simulations is low enough not to affect the model accuracy.

Finally, this research was conducted with the extensive use of DAKOTA toolkit. It was flexible enough to be adapted for several different tests and applications. Moreover, it allowed an easy coupling with an extensive list of other tools. Thanks to its restart capabilities it also enabled secure but controlled, parallel CFD computations. Furthermore, most importantly, one of its most significant advantages was it characteristic of exporting data in very convenient formats and thus enable extensive and automatable visualisation analysis.

10.1 Results and discussion

The routine thus constructed is tested with two design applications from the marine field: a Hull Vane and an aft hull shape optimisation. The goal in both cases is to obtain a reliable resistance reduction

through shape modifications, to do it automatically, and to keep time and costs as low as possible when doing so.

In the first test, the scope was to optimise the position and the shape of a Hull Vane installed at the stern of a ferry. Three variables were used to describe the allowed geometries variations, namely chord length, x and z position. The second test was slightly more complicated and involved the optimisation of the aft shape of an inland vessel sailing in shallow waters. Four variables were used to modify the geometry.

For both tests, CFD settings and SBO settings are described in the design application part of this thesis. Here we only focus on the results and their discussion. At the cost of 35 simulations for the former, and 60 for the latter, both had outstanding performances. The main reasons for that are:

- Both found a global optimum where resistance was reduced with respect to the benchmark. The Hull Vane test provided a reduction of -19.39% for the benchmark. Furthermore, a local minimum was able to reduce the resistance of -18.97%. The new aft shape, instead, gave a reduction of almost 7%.
- 19 (and 27) points of the Halton sequence can uniformly cover a three (and four) dimensions design space. Furthermore, their position is a good starting point for the surrogate model.
- Useful visualisation tools and mid-routine checks were developed to catch trends of resistance and possible anomalies. This allowed a more robust refinement phase.
- The ratio between initial and refinement points proved to ensure a well balanced exploration-exploitation search.
- Choosing Kriging method, and therefore allowing EI function to select new points, was a winning choice. Moreover, this metamodel proved to be flexible but precise enough.

The Hull Vane SBO was not just successful; it was also validated against old results on the same geometry performed by Van Oossanen. These results are a fair comparison because the best Hull Vane has been installed on the ferry. Speed trials were performed before and after the installation of the Hull Vane and a reduction of 19% of shaft power was experienced when sailing at design speed.

The aft ship SBO was an excellent example to show possible improvements to the routine. In particular, we explored the possibility to add a displacement constraint to the resistance reduction problem. We showed that a correct way to implement this feature could be the use of a two-objective optimisation. This way, a trade-off between resistance reduction and displacement increase can be achieved.

In conclusion, two are the reasons why this new routine is an improvement for the old one:

- The new proportion between initial and refinement points allowed a better balance between exploration and exploitation of the design space. The new SBO found the same point of local minimum where the old optimisation stopped, and moved on to a global optimum with lower resistance. This allows the Naval

Architect to have more freedom and maybe to choose between different designs based on more constraints or objectives.

- The new proposed visualisation procedure forces the user to stop and critically analyse the results before starting with the refinement runs. This system does not diminish the automatism of the routine. On the contrary, it provides two new advantages. First, the user must look at the results obtained so far and, doing so, inevitably gains more knowledge on the problem. This awareness will allow him/her in the future to set up an improved version of the SBO. Second, by inspecting the initial results, any mistake, any irregularity is spotted and thus corrected. This means that the accuracy of the method increases exponentially.

10.2 Further improvements

Finally, we present here some topics that could represent further improvement and research for what is covered in this document. If we look back at Figure 2.1, there are three steps of the Surrogate Optimisation workflow that this thesis did not take into account. They are parameter generation, shape deformation, and mesh and CFD simulations. The first two are deeply related to the performance of a SBO and could be further developed, while the third one would require much more work and falls into a different research topic.

First, in order to obtain a successful optimisation, a major role is occupied by the variable selection. How many variables and which ones are sufficient and necessary to fully represent a problem? And equally important, what kind of boundaries to set on these variables?

Furthermore, a good SBO is based on the automate generation of parametric geometries. Which software tools are better for this? What is the more robust way to deal with this issue?

Nomenclature

The next list describes several symbols that were used within the body of the document.

Δ	Distance between analytical and approximated best solutions
δ_0	Initial step length in Pattern Search
ϵ %	Percent relative error
ϵ_{stop}	Stopping tolerance (PS)
λ	Step size in Quasi-Newton search
$\mu_{a,w}$	Air or water dynamic viscosity
Φ	Cumulative distribution function
ϕ	Probability density function
Ψ	Activation function (ANN). Gram Matrix (RBF). Correlation matrix (KG).
ψ	Basis functions
$\rho_{a,w}$	Air or water density
σ^2	Variance
θ, p	Hyper-parameters of KG
φ	Radical inverse function for Halton sequence
$\boldsymbol{B}$	Hessian matrix
$\boldsymbol{C}$	Covariance matrix
$\boldsymbol{X}$	Sampling plan
$\boldsymbol{x}^{(i)}$	Discrete observations or samples
$\boldsymbol{Z}$	Set of random vectors for KG
$\widehat{\mu}$	MLE mean
$\widehat{f}$	Surrogate (approximated) surface
$\widehat{y}$	Approximated resistance for any point on the surrogate surface
c	Contractor factor for Pattern Search
$c^{(i)}$	i sample point position for RBF
D	Design space, or design domain

d	Number of design variables
d_p	Euclidean distance when $p = 2$
e_i	Standard basis for a i=1,2,...,n dimensional space
o	Observed value by DAKOTA for point $\boldsymbol{x}^{(i)}$
p	Surrogate model prediction by DAKOTA for point $\boldsymbol{x}^{(i)}$
s^2	Mean square error in a Gaussian process
w	Weights for basis functions
y	True resistance for a sample point
$y_{\min}$	Best observed data by surrogate
f	"Black Box" function to be approximated
$\boldsymbol{C_\Lambda}$	Corrected covariance matrix
A, θ	Matrix and vector of neuron weights and offset values in ANN
E	Expectation
E[I]	Expected Improvement
Fr	Froude number
I	Improvement
L	Likelihood function
P[I]	Probability of improvement
Rp	Pressure resistance
Rt	Total resistance
Rt %	Percentage of resistance reduction (if negative) or increase (if positive) with respect to benckmark
Rv	Viscous resistance

List of Terms

ANN Artificial Neural Networks. 33, 36, 63
AS Adaptive Sampling. 62

BFGS Broyden Fletcher Gordfarb Shanno. 15, 24, 48

CAD Computer Aided Design. 4, 14
CAE Computer Aided Engineering. 14
CFD Computational Fluid Dynamics. 3, 4, 13
CI Confidence Interval. 73
CV Cross-Validation. 38

DAKOTA Design and Analysis toolKit for Optimisation and Terascale Applications. 5
DIRECT DIvision of RECTangles. 59
DNS Direct Numerical Simulations. 14

EGO Efficient Global Optimisation. 59
EI Expected Improvement. 15, 30, 56, 58

FFNN Feed-Forward Neural Network. 64

GA Genetic Algorithm. 23, 45

HV Hull Vane. 96

IACC International America's Cup Class. 17

KG Kriging. 33, 36, 69

LES Large Eddy Simulations. 14
LHS Latin Hypercube Sampling. 53
LOO Leave-One-Out. 38

MARIN MAritime Research Institute Netherlands. 18
MARS Multivariate Adaptive Regression Splines. 33
MaxAE Maximum Absolute Error. 38
MDO Multi-Disciplinary Optimisation. 21
MLE Maximum Likelihood Estimation. 71, 75
MLP Multi-Layer Perceptron. 64
MLS Moving-Least Square. 33
MOGA Multi-Objective Genetic Algorithm. 24
MOO Multi-Objective Optimisation. 21
MSE Mean Square Error. 30, 72

PoI Probability of Improvement. 30, 57
PR Polynomial Regression. 33
PRESS PRediction Error Sum of Squares. 38
PS Pattern Search. 50
PSO Particle Swan Optimisation. 24

QN Quasi Newton. 48

RANS Reynolds Averaged Navier Stokes. 14

RBF Radial Basis Functions. 33, 36, 66
RMP Race Modelling Program. 19
RMSE Root Mean Square Error. 37

SBG Surrogate Based Global. 61
SBL Surrogate Based Local. 61
SBO Surrogate Based Optimisation. 3, 15
SLP Stochastic Layered Perceptron. 65
SVM Support Vector Machines. 33

UD Uniform Design. 19
UniGe Universitá degli studi di Genova. 18

VPP Velocity Prediction Program. 17–19

Bibliography

Here are the references in author alphabetical order.

[1] E. Acar, 'Effect of error metrics on optimum weight factor selection for ensemble of metamodels', *Expert Systems with Applications*, vol. 42, pp. 2703–2709, Apr. 2015. DOI: 10.1016/j.eswa.2014.11.020 (cited on pages 37, 79).

[2] B. M. Adams, M. S. Eldred, G. Geraci, R. W. Hooper, J. D. Jakeman, K. A. Maupin, J. A. Monschke, A. A. Rushdi, J. A. Stephens, L. P. Swiler, and T. M. Wildey, 'Dakota, a multilevel parallel object-oriented framework for design optimization, parameter estimation, uncertainty quantification, and sensitivity analysis: Version 6.10 developers manual', Nov. 2019 (cited on page 5).

[3] ——, 'Dakota, a multilevel parallel object-oriented framework for design optimization, parameter estimation, uncertainty quantification, and sensitivity analysis: Version 6.10 reference manual', May 2019 (cited on pages 5, 55, 66, 69, 74, 91).

[4] ——, 'Dakota, a multilevel parallel object-oriented framework for design optimization, parameter estimation, uncertainty quantification, and sensitivity analysis: Version 6.10 theory manual', Nov. 2019 (cited on pages 5, 59).

[5] ——, 'Dakota, a multilevel parallel object-oriented framework for design optimization, parameter estimation, uncertainty quantification, and sensitivity analysis: Version 6.10 user's manual', May 2019 (cited on pages 5, 6, 23–25, 37, 38, 59, 61, 66, 74, 79–81, 83, 91).

[6] D. Andrews, 'Creative ship design', 1981 (cited on page 12).

[7] C. Audet and W. Hare, 'Derivative-free and blackbox optimization', Jan. 2017. DOI: 10.1007/978-3-319-68913-5 (cited on pages 21–24, 45, 46, 48, 50, 54, 77).

[8] J. F. M. Barthelemy and R. T. Haftka, 'Approximation concepts for optimum structural design: A review', *Structural optimization*, vol. 5, no. 3, pp. 129–144, Sep. 1993. DOI: 10.1007/BF01743349 (cited on pages 17, 32, 64).

[9] C. G. Broyden, 'Quasi-newton methods and their application to function minimisation', *Mathematics of Computation*, vol. 21, pp. 368–381, 1967 (cited on page 23).

[10] I. Buxton and B. S. R. Association, *Engineering Economics and Ship Design*. British Ship Research Association, 1976 (cited on page 12).

[11] R. H. Byrd, R. B. Schnabel, and G. A. Shultz, 'Parallel quasi-newton methods for unconstrained optimization', *Math. Program.*, vol. 42, no. 2, pp. 273–306, Sep. 1988. DOI: 10.1007/BF01589407 (cited on page 49).

[12] V. Chen, K.-L. Tsui, R. Barton, and M. Meckesheimer, 'A review on design, modeling and applications of computer experiments', *Iie Transactions*, vol. 38, pp. 273–291, Apr. 2006. DOI: 10.1080/07408170500232495 (cited on page 28).

[13] B. Cointe, 'Optimisation de carène par modèle de substitution', Ecole Centrale de Nantes, Aug. 2016 (cited on pages 15, 18).

[14] A. Cominetti, 'Open-source shape optimization: An application to bulbous bow', Universitá degli Studi di Genova, Mar. 2017 (cited on pages 18, 34).

[15] P. Couser, A. Mason, G. Mason, C. Smith, and B. von Konsky, 'Artificial neural networks for hull resistance prediction', Apr. 2004 (cited on pages 17, 18).

[16] Q. Delivré, 'Optimisation de carène par modèle de substitution', Ecole Centrale de Nantes, Sep. 2014 (cited on pages 15, 16, 18, 28, 31, 34, 35, 79).

[17] J. E. Dennis and B. Schnabel, 'Numerical methods for unconstrained optimization and nonlinear equations', in *Prentice Hall series in computational mathematics*, 1983 (cited on page 49).

[18] N. Dyn, D. Levin, and S. Rippa, 'Numerical procedures for surface fitting of scattered data by radial functions', *SIAM Journal on Scientific and Statistical Computing*, vol. 7, Apr. 1986. DOI: 10.1137/0907043 (cited on page 66).

[19] A. E. Eiben and C. A. Schippers, 'On evolutionary exploration and exploitation', *Fundam. Inf.*, vol. 35, no. 1–4, pp. 35–50, Aug. 1998 (cited on page 46).

[20] J. H. Evans, 'Basic design concepts', *Journal of the American Society for Naval Engineers*, vol. 71, no. 4, pp. 671–678, 1959. DOI: 10.1111/j.1559-3584.1959.tb01836.x (cited on page 12).

[21] K. T. Fang, D. K. Lin, P. Winker, and Y. Zhang, 'Uniform design: Theory and application', *Technometrics*, vol. 42, no. 3, pp. 237–248, Aug. 2000. DOI: 10.1080/00401706.2000.10486045 (cited on page 27).

[22] A. Forrester and A. Keane, 'Recent advances in surrogate-based optimization', *Progress in Aerospace Sciences*, vol. 45, pp. 50–79, Apr. 2009. DOI: 10.1016/j.paerosci.2008.11.001 (cited on pages 67, 68, 79).

[23] A. Forrester, A. Sobester, and A. Keane, *Engineering Design Via Surrogate Modelling: A Practical Guide*. Wiley, Jul. 2008 (cited on pages 26, 29, 30, 32, 57–59, 66, 68, 70–72, 79).

[24] D. E. Goldberg, 'Genetic algorithms in search optimization and machine learning', 1988 (cited on pages 23, 46).

[25] J. Guerrero, 'An open-source framework for multi-physics simulations and optimization', Mar. 2015 (cited on page 18).

[26] ——, 'Opportunities and challenges in cfd optimization: Open source technology and the cloud', Jun. 2018. DOI: 10.13140/RG.2.2.25427.02086 (cited on page 38).

[27] J. Guerrero, G. Bailardi, H. Telib, and R. Lyulinetskyy, 'An open-source framework for multi-physics simulations, design space exploration and design optimization', Jun. 2015 (cited on page 18).

[28] J. Guerrero, A. Cominetti, J. Pralits, and D. Villa, 'Surrogate-based optimization using an open-source framework: The bulbous bow shape optimization case', *Mathematical and Computational Applications*, vol. 23, p. 60, Oct. 2018. DOI: 10.3390/mca23040060 (cited on pages 18, 22, 24, 28, 31, 33, 38, 39, 79).

[29] R. T. Haftka and Z. Gürdal, *Elements of Structural Optimization*. Springer, 1992 (cited on page 46).

[30] P. Hajela and L. Berke, 'Neural networks in structural analysis and design: An overview', *Computing Systems in Engineering*, vol. 3, no. 1, pp. 525–538, 1992, High-Performance Computing for Flight Vehicles. DOI: https://doi.org/10.1016/0956-0521(92)90138-9 (cited on page 64).

[31] J. H. Halton, 'On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals', *Numerische Mathematik*, vol. 2, pp. 84–90, 1960 (cited on pages 27, 55).

[32] J. M. Hammersley, 'Monte carlo methods for solving multivariable problems', *Annals of the New York Academy of Sciences*, vol. 86, no. 3, pp. 844–874, 1960. DOI: 10.1111/j.1749-6632.1960.tb42846.x (cited on page 27).

[33] R. Hardy, 'Multivariate equations of topography and other irregular surfaces', *Journal of Geophysical Research*, vol. 71, pp. 1905–1915, Jan. 1971 (cited on page 66).

[34] A. Hedayat, N. Sloane, and J. Stufken, 'Orthogonal arrays: Theory and applications', *Technometrics*, vol. 42, Nov. 2000. DOI: 10.2307/1270971 (cited on page 27).

[35] R. R. Hilleary, 'The tangent search method of constrained minimization', Monterey, California. Naval Postgraduate School, Tech. Rep., 1966 (cited on page 24).

[36] J. H. Holland, 'Adaptation in natural and artificial systems', 1975 (cited on page 46).

[37] J. Holtrop and G. Mennen, 'An approximate power prediction method', 1982 (cited on page 123).

[38] R. Hooke and T. A. Jeeves, '"direct search" solution of numerical and statistical problems', *J. ACM*, vol. 8, no. 2, pp. 212–229, Apr. 1961. DOI: 10.1145/321062.321069 (cited on page 24).

[39] E. Iuliano, 'Efficient design optimization assisted by sequential surrogate models', *International Journal of Aerospace Engineering*, vol. 2019, p. 34, May 2019. DOI: 10.1155/2019/4937261 (cited on pages 30, 32, 79).

[40] E. Iuliano and E. A. Perez, *Application of Surrogate-based Global Optimization to Aerodynamic Design*. Springer, 2016 (cited on pages 31, 79).

[41] E. Iuliano and D. Quagliarella, 'Proper orthogonal decomposition, surrogate modelling and evolutionary optimization in aerodynamic design', *Computers and Fluids*, vol. 84, pp. 327–350, 2013. DOI: https://doi.org/10.1016/j.compfluid.2013.06.007 (cited on pages 18, 31).

[42] A. Jareteg, 'Coupling of dakota and openfoam for automatic parameterized optimization', Chalmers University, Oct. 2014 (cited on page 18).

[43] P. Jiang, Q. Zhou, and X. Shao, 'Surrogate model-based engineering design and optimization', *ASME*, 2020 (cited on pages 29, 30, 36, 68, 74).

[44] R. Jin, W. Chen, and T. Simpson, 'Comparative studies of metamodeling techniques under multiple modeling criteria', *Structural and Multidisciplinary Optimization*, vol. 23, pp. 1–13, Jan. 2001. DOI: 10.1007/s00158-001-0160-4 (cited on pages 27, 29, 33, 36, 55, 68, 74).

[45] R. Jin, W. Chen, and A. Sudjitanto, 'On sequential sampling for global metamodeling in engineering design', vol. 2, Jan. 2002. DOI: 10.1115/DETC2002/DAC-34092 (cited on pages 30, 32, 79).

[46] D. Jones, 'A taxonomy of global optimization methods based on response surfaces', *J. of Global Optimization*, vol. 21, pp. 345–383, Dec. 2001. DOI: 10.1023/A:1012771025575 (cited on pages 29–31, 73, 79).

[47] D. Jones, M. Schonlau, and W. Welch, 'Efficient global optimization of expensive black-box functions', *Journal of Global Optimization*, vol. 13, pp. 455–492, Jun. 1998 (cited on pages 27, 41, 59, 69, 79).

[48] J. Kennedy and R. Eberhart, 'Particle swarm optimization', in *Proceedings of ICNN'95 - International Conference on Neural Networks*, vol. 4, 1995, 1942–1948 vol.4 (cited on page 24).

[49] J. Koehler and A. Owen, '9 computer experiments', *Handbook of Statistics*, vol. 13, pp. 261–308, Dec. 1996. DOI: 10.1016/S0169-7161(96)13011-X (cited on pages 72, 73).

[50] S. Koziel and L. Leifsson, *Surrogate-Based Modeling and Optimization: Applications in Engineering*. Aug. 2013 (cited on pages 23, 46).

[51] D. Krige, 'A statistical approach to some mine valuation and allied problems on the witwatersrand', Mar. 1951 (cited on page 69).

[52] S. Larson, 'The shrinkage of the coefficient of multiple correlation', *Educat. Psychol*, 1931 (cited on page 38).

[53] L. Larsson and H. C. Raven, *Principles of Naval Architecture Series - Ship Resistance and Flow*. Society of Naval Architects and Marine Engineers (SNAME), 2010 (cited on page 13).

[54] P. Le Bihan, 'User manual optimizer', Van Oossanen Naval Architects, Feb. 2018 (cited on pages 15, 27, 41).

[55] A. Mason, 'Stochastic optimisation for america's cup class yachts based on a genetic algorithm', *Schiffstechnick*, vol. 55, p. 60, Jan. 2008 (cited on page 18).

[56] ——, 'Stochastic optimisation of america's cup class yachts', PhD thesis, Australian Maritime College University of Tasmania, Feb. 2010 (cited on pages 18, 32–34, 64, 65).

[57] A. Mason, P. Couser, G. Mason, C. Smith, and B. von Konsky, 'Optimisation of vessel resistance using genetic algorithms and artificial neural networks', *COMPIT*, Jan. 2005 (cited on pages 17, 22, 24, 64).

[58] A. Mason and G. Thomas, 'Stochastic optimisation of iacc yachts', in *COMPIT 2007, At Cortona, Italy*, Apr. 2007 (cited on page 18).

[59] D. Mavris and D. DeLaurentis, 'Methodology for examining the simultaneous impact of requirements, vehicle characteristics, and technologies on military aircraft design', in *Proceedings of the 22st Congress of the ICAS*, Jan. 2000 (cited on page 12).

[60] M. D. McKay, R. J. Beckman, and W. J. Conover, 'Comparison of three methods for selecting values of input variables in the analysis of output from a computer code', *Technometrics*, vol. 21, no. 2, pp. 239–245, 1979. DOI: 10.1080/00401706.1979.10489755 (cited on pages 27, 53).

[61] M. Meckesheimer, A. Booker, R. Barton, and T. Simpson, 'Computationally inexpensive metamodel assessment strategies', *AIAA Journal*, vol. 40, pp. 2053–2060, Oct. 2002. DOI: 10.2514/2.1538 (cited on page 37).

[62] C. A. Micchelli, 'Interpolation of scattered data: Distance matrices and conditionally positive definite functions', *Constructive Approximation*, Dec. 1986. DOI: https://doi.org/10.1007/BF01893414 (cited on page 68).

[63] J. A. Nelder and R. Mead, 'A simplex method for function minimization', *Comput. J.*, vol. 7, pp. 308–313, 1965 (cited on page 24).

[64] A. T. Nguyen, S. Reiter, and P. Rigo, 'A review on simulation-based optimization methods applied to building performance analysis', *Applied Energy*, vol. 113, Jan. 2014. DOI: 10.1016/j.apenergy.2013.08.061 (cited on pages 22, 23).

[65] A. Ohm and H. Ö. Pétursson, 'Automated cfd optimisation of a small hydro turbine forwater distribution networks', 2017 (cited on page 18).

[66] M. J. L. Orr, 'Introduction to radial basis function networks', University of Edinburgh, 1996 (cited on pages 38, 69).

[67] A. B. Owen, 'Orthogonal arrays for computer experiments, integration and visualization', 1992 (cited on page 27).

[68] J.-S. Park, 'Optimal latin-hypercube designs for computer experiments', *Journal of Statistical Planning and Inference*, vol. 39, no. 1, pp. 95–111, 1994. DOI: https://doi.org/10.1016/0378-3758(94)90115-5 (cited on page 27).

[69] D. Peri and E. Campana, 'Multidisciplinary design optimization of a naval surface combatant', *Journal of Ship Research*, vol. 47, pp. 1–12, Mar. 2003 (cited on page 22).

[70] V. Picheny, 'Improving accuracy and compensating for uncertainty in surrogate modeling', Dec. 2009 (cited on pages 36, 73, 79).

[71] N. Queipo, R. Haftka, W. Shyy, T. Goel, R. Vaidyanathan, and P. Tucker, 'Surrogate-based analysis and optimization', *Progress in Aerospace Sciences*, vol. 41, pp. 1–28, Jan. 2005. DOI: 10.1016/j.paerosci.2005.02.001 (cited on pages 34, 68).

[72] H. Raven and T. Scholcz, 'Wave resistance minimisation in practical ship design', May 2017 (cited on pages 18, 24, 28).

[73] H. Raven and T. Scholcz, 'An assessment of multifidelity procedures for ship hull form optimization', May 2019 (cited on pages 18, 28, 33, 79).

[74] S. Razavi, B. Tolson, and D. Burn, 'Review of surrogate modeling in water resources', *Water Resources Research*, vol. 48, pp. 7401–, Jul. 2012. DOI: 10.1029/2011WR011527 (cited on pages 64, 65, 68, 74).

[75] H. H. Rosenbrock, 'An Automatic Method for Finding the Greatest or Least Value of a Function', *The Computer Journal*, vol. 3, no. 3, pp. 175–184, Jan. 1960. DOI: 10.1093/comjnl/3.3.175 (cited on page 78).

[76] R. A. Rota Roselli, G. Vernengo, C. Altomare, S. Brizzolara, L. Bonfiglio, and R. Guercio, 'Ensuring numerical stability of wave propagation by tuning model parameters using genetic algorithms and response surface methods', *Environmental Modelling and Software*, vol. 103, pp. 62–73, 2018. DOI: https://doi.org/10.1016/j.envsoft.2018.02.003 (cited on page 18).

[77] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, 'Learning representations by back-propagating errors', *Nature*, vol. 323, pp. 533–536, 1986 (cited on page 65).

[78] J. Sacks, S. Schiller, and W. Welch, 'Designs for computer experiments', *Technometrics*, vol. 31, pp. 41–47, Feb. 1989. DOI: 10.1080/00401706.1989.10488474 (cited on page 69).

[79] J. Sacks, W. Welch, T. Mitchell, and H. Wynn, 'Design and analysis of computer experiments', *Statistical Science*, vol. 4, Jan. 1989. DOI: 10.1214/ss/1177012413 (cited on pages 27, 32, 37, 69, 70, 72).

[80] W. S. Sarle, 'Neural networks and statistical models', *Nineteenth Annual SAS Users Group International Conference*, 1994 (cited on page 65).

[81] T. Scholcz and E. Daalen, 'Surrogate-based multi-objective optimisation for powering and seakeeping', Sep. 2019 (cited on pages 18, 24).

[82] T. Scholcz, T. Gornicz, and C. Veldhuis, 'Multi-objective hull-form optimization using kriging on noisy computer experiments', Jun. 2015. DOI: 10.13140/RG.2.1.4173.5124 (cited on page 34).

[83] T. Scholcz and C. Veldhuis, 'Multi-objective surrogate based hull-form optimization using high-fidelity rans computations', May 2017 (cited on page 31).

[84] L. Shu, P. Jiang, Q. Zhou, L. Wan, X. Shao, and Y. Zhang, 'Metamodel-based design optimization employing a novel sequential sampling strategy', *Engineering Computations*, vol. 34, Feb. 2018. DOI: 10.1108/EC-01-2016-0034 (cited on pages 37, 38).

[85] T. Simpson, T. Mauery, J. Korte, and F. Mistree, 'Comparison of response surface and kriging models for multidisciplinary design optimization', vol. 98, Sep. 1998 (cited on page 70).

[86] T. Simpson, J. Poplinski, P. Koch, and J. Allen, 'Metamodels for computer-based engineering design: Survey and recommendations', *Engineering with Computers*, vol. 17, pp. 129–150, Jul. 2001. DOI: 10.1007/PL00007198 (cited on pages 26, 28, 34, 64, 69, 74).

[87] A. Sobester and A. Forrester, 'Aircraft aerodynamic design: Geometry and optimization', Nov. 2014 (cited on page 48).

[88] A. Sobester, S. Leary, and A. Keane, 'On the design of optimization strategies based on global response surface approximation models', *Journal of Global Optimization*, vol. 33, pp. 31–59, Sep. 2005. DOI: 10.1007/s10898-004-6733-1 (cited on pages 29, 41, 79, 89).

[89] I. Spisso, 'Parametric and optimization study: Openfoam and dakota', Nov. 2012 (cited on page 18).

[90] M. Stone, 'Cross-validatory choice and assessment of statistical predictions', *Royal Stat. Soc.*, 1974 (cited on page 38).

[91] K. Uithof, N. Hagemeister, B. Bouckaert, P. van Oossanen, and N. Moerke, 'A systematic comparison of the influence of the hull vane®, interceptors, trim wedges, and ballasting on the performace of the 50m amecrc series# 13 patrol vessel', *Proc. Warship*, 2016 (cited on page 96).

[92] S. Ulaganathan and N. Asproulis, *Surrogate Models for Aerodynamic Shape Optimisation*. Springer, 2013 (cited on page 18).

[93] V. N. Vapnik, *Statistical Learning Theory*, 1. edition, ser. Adaptive and Learning Systems for Signal Processing, Communications, and Control. Wiley-Interscience, Sep. 1998 (cited on page 68).

[94] F. Vesting and R. Bensow, 'On surrogate methods in propeller optimisation', *Ocean Engineering*, vol. 88, pp. 214–227, Sep. 2014. DOI: 10.1016/j.oceaneng.2014.06.024 (cited on pages 18, 24).

[95] F. Viana, R. Haftka, and V. Steffen Jr, 'Multiple surrogates: How cross-validation errors can help us to obtain the best predictor', *Structural and Multidisciplinary Optimization*, vol. 39, pp. 439–457, Oct. 2009. DOI: 10.1007/s00158-008-0338-0 (cited on page 79).

[96] F. Viana, T. Simpson, V. Balabanov, and V. Toropov, 'Metamodeling in multidisciplinary design optimization: How far have we really come?', *AIAA Journal*, vol. 52, Mar. 2014. DOI: 10.2514/1.J052375 (cited on pages 4, 5, 34).

[97] G. Wang and S. Shan, 'Review of metamodeling techniques in support of engineering design optimization', *Journal of Mechanical Design - J MECH DESIGN*, vol. 129, Apr. 2007. DOI: 10.1115/1.2429697 (cited on pages 26, 27, 34, 37, 38, 54, 56).

[98] R. Whitfield, P. WRIGHT, G. Coates, and W. HILLS, 'A robust design methodology suitable for application to one-off products', *Journal of Engineering Design - J ENGINEERING DESIGN*, vol. 9, pp. 373–387, Dec. 1998. DOI: 10.1080/095448298261507 (cited on page 24).

[99] R. Wilson, F. Stern, H. Coleman, and E. Paterson, 'Comprehensive approach to verification and validation of cfd simulations—part 2: Application for rans simulation of a cargo/container ship', *Journal of Fluids Engineering-transactions of The Asme - J FLUID ENG*, vol. 123, Dec. 2001. DOI: 10.1115/1.1412236 (cited on page 27).

[100] X.-S. Yang, *Engineering Optimization: An Introduction with Metaheuristic Applications*. Sep. 2010 (cited on page 23).

[101] K. S. Zaaijer, 'Aft ship optimization: A means to an end', TU Delft, Aug. 2009 (cited on page 15).

[102] P. Zadeh, V. Toropov, and A. Wood, 'Metamodel-based collaborative optimization framework', *Structural and Multidisciplinary Optimization*, vol. 38, pp. 103–115, Apr. 2009. DOI: 10.1007/s00158-008-0286-8 (cited on pages 26, 36).

[103] D. Zhao and D. Xue, 'A comparative study of metamodeling methods considering sample quality merits', *Structural and Multidisciplinary Optimization*, vol. 42, pp. 923–938, Dec. 2010. DOI: 10.1007/s00158-010-0529-3 (cited on pages 37, 38).

[104] D. C. Zimmerman, 'Genetic algorithms for navigating expensive and complex design spaces', *Technical report, Final Report for Sandia NationalLaboratories*, vol. contract AO-7736 CA 02, Sep. 1996 (cited on page 65).