CFD BASED AERODYNAMIC DESIGN OPTIMIZATION USING BAYESIAN INFERENCE AND KRIGING SURROGATE MODEL

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

YUNUS EMRE SUNAY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
AEROSPACE ENGINEERING

JANUARY 2024

Approval of the thesis:

CFD BASED AERODYNAMIC DESIGN OPTIMIZATION USING BAYESIAN INFERENCE AND KRIGING SURROGATE MODEL

submitted by **YUNUS EMRE SUNAY** in partial fulfillment of the requirements for the degree of **Master of Science in Aerospace Engineering, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, **Graduate School of Natural and Applied Sciences** _____

Prof. Dr. Serkan Özgen
Head of the Department, **Aerospace Engineering** _____

Assoc. Prof. Dr. Nilay Sezer Uzol
Supervisor, **Aerospace Engineering, METU** _____

Examining Committee Members:

Prof. Dr. Yusuf Özyörük
Aerospace Engineering Department, METU _____

Assoc. Prof. Dr. Nilay Sezer Uzol
Aerospace Engineering Department, METU _____

Prof. Dr. Dilek Funda Kurtuluş
Aerospace Engineering Department, METU _____

Asst. Prof. Dr. Özge Başkan Perçin
Aerospace Engineering Department, METU _____

Asst. Prof. Dr. Onur Baş
Mechanical Engineering Department, TEDU _____

Date: 18.01.2024

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name Last name : YUNUS EMRE SUNAY

Signature :

ABSTRACT

CFD BASED AERODYNAMIC DESIGN OPTIMIZATION USING BAYESIAN INFERENCE AND KRIGING SURROGATE MODEL

Sunay, Yunus Emre
Master of Science, Aerospace Engineering
Supervisor : Assoc. Prof. Dr. Nilay Sezer Uzol

January 2024, 90 pages

The advancement of computer hardware technology significantly contributes to resolving complex engineering challenges that require solution of complex, highly nonlinear equations. The fidelity of these analyses is closely tied to the capabilities of computing systems and improves with advancements in computing power. However, even with high-fidelity numerical simulators, obtaining a solution can be time-consuming, often requiring several hours. This study aims to integrate high-fidelity analytical tools into the design cycle and thereby to reduce the overall duration of the design process through enhanced computational power. An optimization tool based on Bayesian inference process is developed to find the optimum design. In this study, deterministic numerical simulations are employed to construct a Kriging surrogate model, which represents a stochastic relationship between inputs and outputs, independent of the underlying physics of the simulations. Computational Fluid Dynamics, an expensive high-fidelity tool, is replaced by the Kriging surrogate model in Bayesian optimization. Latin Hypercube Sampling, which has randomness and homogeneity, is used to initialize the samples in the design space; however, it is not possible to increase the number of samples for

iterative processes. Furthermore, this study investigates a hybrid methodology for parallel incremental sampling, which uses randomness and even distribution of LHS. This method is applicable for any initial quantity of samples to target number of samples. The optimization tool, which contains hybrid parallel algorithms for sampling, is tested with well-known Rosenbrock function and applied for shape optimization of delta wing using meshless CFD.

Keywords: Kriging Surrogate Model, Bayesian Optimization, Meshless Computational Fluid Dynamics, Vortex Particle Method, Parallel Design Strategies

ÖZ

HAD TABANLI AERODİNAMİK TASARIM OPTİMİZASYONDA BAYESIAN ÇIKARIM VE KRIGING IKAME MODEL KULLANILMASI

Sunay, Yunus Emre
Yüksek Lisans, Havacılık ve Uzay Mühendisliği
Tez Yöneticisi: Doç. Dr. Nilay Sezer Uzol

Ocak 2024, 90 sayfa

Bilgisayar donanım teknolojisinin gelişimi, karmaşık ve doğrusal olmayan denklemlerin çözümünü gerektiren mühendislik problemlerinin çözümüne önemli ölçüde katkı sağlamıştır. Bu analizlerin gerçeğe uygunluğu, hesaplama sistemlerinin yeteneklerine yakından bağlıdır ve hesaplama gücündeki gelişmelerle birlikte gelişmektedir. Ancak, yüksek doğruluklu sayısal simülatörlerle bir çözüm elde etmek bile zaman alıcı olmaktadır ve genellikle saatler mertebesinde sürmektedir. Bu çalışmada, yüksek doğruluklu analitik araçların tasarım döngüsüne entegre edilmesi ve bilgisayar hesaplama gücü kullanarak tasarım sürecinin toplam süresinin azaltılması amaçlanmaktadır. Optimum tasarımı bulmak amacıyla Bayesian çıkarım sürecine dayalı bir optimizasyon aracı geliştirilmiştir. Bu çalışmada, deterministik sayısal simülasyonlar kullanarak girdi/çıktı arasındaki stokastik ilişkiyi temsil eden Kriging ikame model oluşturulmuştur. Pahalı yüksek doğruluğa sahip hesaplama yöntemi olan Hesaplamalı Akışkanlar Dinamiği, Bayesian optimizasyonunda Kriging ikame modeli ile modellenmiştir. Rastgelelik ve homojenliğe sahip Latin Hiperküp Örneklemesi, tasarım uzayındaki ilk örneklemeleri oluşturmak için kullanılmaktır; ancak iteratif döngülerde örnek sayısını artırmak mümkün değildir.

Ayrıca bu çalışma, rastgelelik ve Latin Hiperküp Örneklemesinin homojen dağılımını kullanan paralel artan örnekleme için hibrit bir metodolojiyi araştırmaktadır. Bu yöntem herhangi bir ilk örnek sayısından istenilen hedef örnek sayılarına uygulanabilmektedir. Örnekleme için hibrit paralel örnekleme algoritmaları içeren optimizasyon aracı, optimizasyon test fonksiyonu Rosenbrock fonksiyonuyla test edilmiş ve çözüm ağsız CFD kullanarak delta kanat şekil optimizasyonuna uygulanmıştır.

Anahtar Kelimeler: Kriging İkame Model, Bayesian Optimizasyonu, Çözüm Ağsız Hesaplamalı Akışkanlar Dinamiği, Girdap Parçacık Metodu, Paralel Tasarım Stratejileri

To my cats

ACKNOWLEDGMENTS

The author wishes to express his deepest gratitude to his supervisor Assoc. Prof. Dr. Nilay Sezer Uzol for their guidance, advice, criticism, encouragements and insight throughout the research.

The author would also like to thank his wife Şermin Saatçıoğlu for her suggestions, comments and help.

The author would also like to thank his family for their support.

TABLE OF CONTENTS

ABSTRACT v

ÖZ vii

ACKNOWLEDGMENTS x

TABLE OF CONTENTS xi

LIST OF TABLES xiii

LIST OF FIGURES xiv

LIST OF ABBREVIATIONS xvii

CHAPTERS

1 INTRODUCTION 1

1.1 Literature Review 5

1.1.1 Design of Experiment (DoE) 6

1.1.2 Kriging Surrogate Model (Gaussian Process Regression) and Bayesian Optimization 8

1.2 Aim of the Thesis 13

1.3 Outline of the Thesis 14

2 METHODOLOGY 17

2.1 Rosenbrock Test Function 20

2.2 Sampling Algorithms/Strategies 21

2.2.1 Latin Hypercube Sampling (LHS) 21

2.2.2 Hybrid Incremental Sampling Method 22

2.3 The Kriging Surrogate Model 26

2.4 Leave-One-Out Cross-Validation (LOOCV) 30

2.5 Bayesian Optimization 32
2.5.1 Gaussian Process Regression (GPR) 33
2.5.2 Acquisition Function 36
2.6 Unsteady Flow Solver (FLOWUnsteady) 38
2.6.1 Validation of Unsteady Flow Solver (FLOWUnsteady) 44
2.7 Optimization Tool Development 46
3 RESULTS AND DISCUSSION 49
3.1 Surrogate Model Optimization for Rosenbrock Function 50
3.2 Surrogate Model Optimization for Delta Wing 56
3.2.1 Optimization of Delta Wing with Twist Angles 58
3.2.2 Optimization of Delta Wing with Aspect Ratio, Taper Ratio and Sweep Angle 66
4 CONCLUSION 79
REFERENCES 83
APPENDICES 89
A. FLOWUnsteady Unsteady Flow Solver 89

LIST OF TABLES

TABLES

Table 2.1 Wind tunnel flow condition 45
Table 2.2 Comparison of numerical simulation and experimental results..... 45
Table 3.1 Results of surrogate model and numerical simulation and R^2, RMSE (F_m is result of surrogate model, F_r is result of Rosenbrock function) 56
Table 3.2 Comparison of numerical simulation and experimental results..... 57
Table 3.3 Design space of twist angles 58
Table 3.4 Results of surrogate model and numerical simulation and R^2, RMSE (F_m is result of surrogate model, F_{CFD} is result of CFD, θ is twist angle)..... 62
Table 3.5 Comparison result of optimum and baseline design 64
Table 3.6 Design space of optimization of delta wing..... 66
Table 3.7 Results of surrogate model and numerical simulation and R^2, RMSE (F_m: Result of surrogate model, F_{CFD}: Result of CFD, AR: Aspect ratio, TR: Taper ratio, Λ: Sweep angle) 71
Table 3.8 Comparison results of optimum and baseline design (AR: Aspect Ratio, TR: Taper Ratio, Λ: Sweep Angle)..... 73
Table 3.9 Results of surrogate model and numerical simulation, and R^2 and RMSE (F_m: Result of surrogate model, F_{CFD} : Results of CFD, AR: Aspect ratio, TR: Taper ratio, Λ: Sweep angle, $n_{initial}$: number of initial samples)..... 76

LIST OF FIGURES

FIGURES

Figure 1.1 Sample distribution for random samples and LHS samples 3
Figure 1.2 Kriging model in 1D space 4
Figure 1.3 LHS samples (right) and OLHS sample (left) distribution 7
Figure 1.4 Demonstration of lid-driven cavity 9
Figure 1.5 Contours of the velocity magnitude (normalized with U_∞(left) and streamlines of the cavity flow (right) for minimum energy dissipation [36] 10
Figure 1.6 Contours of the velocity magnitude (normalized with U_∞ (left) and streamlines of the cavity flow (right) for maximum energy dissipation [36]..... 10
Figure 1.7 Demonstration of the parameters in shape optimization of the diverging channel [36] 11
Figure 1.8 Results of the constant Clauser PG parameter optimization [36] 11
Figure 1.9 Results of the variable Clauser PG parameter optimization [36]..... 12
Figure 2.1 Flowchart for surrogate model optimization process..... 19
Figure 2.2 Contour of Rosenbrock functions for 2D..... 20
Figure 2.3 Sampling points for the Rosenbrock test function in the design space (left: 20 sampling points, right: 30 sampling points) 22
Figure 2.4 Demonstration of difference between the initial points with LHS and the updated points with the incremental sampling method 25
Figure 2.5 Demonstration of difference between LHS and the hybrid incremental sampling method. 26
Figure 2.6 Kriging surrogate model for 1D..... 30
Figure 2.7 K-fold cross-validation 31
Figure 2.8 Flow regime in the wake of the propeller [48]..... 41
Figure 2.9 Various fidelity levels for FLOWUnsteady solver [48]..... 42
Figure 2.10 Stretching of a spherical fluid element [48]..... 43
Figure 2.11 Vortex tube formed by vortex lines [48]..... 43

Figure 2.12 Geometric parameters of the baseline wing and explanations of the parameters 44
Figure 2.13 Comparison of numerical simulation and experimental results [48]... 45
Figure 2.14 Class scheme of the optimization tool: IncSampMeth_BayesOpt 47
Figure 3.1 Contour of Rosenbrock functions and optimum point 49
Figure 3.2 Initial samples of Rosenbrock functions and optimum point 51
Figure 3.3 Initial samples' results of the Rosenbrock functions in full range 51
Figure 3.4 Initial samples' results of the Rosenbrock functions $f(x) \in 0,100$ 52
Figure 3.5 Variation of RMSE with number of samples 53
Figure 3.6 Distribution of the sample over the iterations 54
Figure 3.7 Comparison results of surrogate model (LOOCV) and Rosenbrock functions 55
Figure 3.8 Flow conditions and geometric parameters of baseline configuration .. 57
Figure 3.9 Demonstration of wing twist angles 58
Figure 3.10 Initial sampling with LHS 59
Figure 3.11 CFD results (C_L/C_D) of initial samples and optimum design 59
Figure 3.12 Initial sampling and increment samples 60
Figure 3.13 RMSE investigated from LOOCV of surrogate model 61
Figure 3.14 Comparison results of CFD and surrogate model (LOOCV) 63
Figure 3.15 C_L and C_D at the optimum configuration 64
Figure 3.16 Spanwise lift and drag coefficient 65
Figure 3.17 Instantaneous wake developments shown with particle distributions in different times 65
Figure 3.18 Demonstration of design parameters 66
Figure 3.19 Initial sampling with LHS 67
Figure 3.20 Different configurations from design space of delta wing 67
Figure 3.21 CFD results (C_L/C_D) of initial samples and optimum design 68
Figure 3.22 Distribution of the samples over the iterations 69
Figure 3.23 RMSE of surrogate model with LOOCV 70
Figure 3.24 Comparison of cross-validation and real functions results 72

Figure 3.25 Baseline (left) and optimum design (right) configurations 73
Figure 3.26 C_L and C_D of the optimum design 74
Figure 3.27 Spanwise lift and drag coefficients of the optimum design 74
Figure 3.28 Instantaneous wake developments shown with particle distributions in different times 75
Figure 3.29 Computational time of optimization cycle 77

LIST OF ABBREVIATIONS

ABBREVIATIONS

BBD	Bos-Behnken Design
CCD	Central Composite Design
CFD	Computational Fluid Dynamics
CPU	Central Processing Unit
DoE	Design of Experiment
EGO	Efficient Global Optimization
EI	Expected Improvement
GPR	Gaussian Process Regression
GPU	Graphics Processing Unit
LES	Large Eddy Simulation
LHS	Latin Hypercube Sampling
MAP	Maximum a Posteriori
MLE	Maximum Likelihood Estimate
OLHS	Optimal Latin Hypercube Sampling
OSFD	Optimal Space Filling Design
PG	Pressure-Gradient
RBF	Radial Basis Function
RSM	Response Surface Methodology
rVPM	reformulated Vortex Particle Methods
TBL	Turbulent Boundary Layers

CHAPTER 1

INTRODUCTION

The development of computer hardware technology is helping to solve very complex engineering problems using solutions of highly nonlinear equations. Fidelity of this analysis depends on the computer abilities and increases with the developments in computer power; on the other hand, it can still take hours to reach a solution for a single case with the supercomputer system. Combination of the size of the design space and the number of variables results in numerous numbers of alternatives in the design space. In addition to that, optimization methods based on differentiation such as gradient descent cannot be parallelized since the successive step depends on the results coming from the previous run. Increasing the number of CPUs can speed up the runtime of the analysis; however, it is still limited by the hardware. All these situations make it impossible for high-fidelity analysis tools to be involved in the design cycle.

The motivation of this study is to involve high fidelity analysis tools in the design cycle and decrease the total time required for the design process with the help of quantity of CPU and GPU and optimization algorithms. This issue can be handled with the help of surrogate model optimization methods. Most remarkable feature of these methods is their ability of parallelization that concludes with decreasing the required design time. Surrogate model optimization consists of three main components which are sampling, building the surrogate model and optimization of the surrogate model. This optimization method can be used as both one-time run and iterative process to obtain the optimum points. The challenging part in the iterative process is how to update samples both keeping initial points and increment samples by parallelization strategy. This study concerns not only adding optimum points to

the design space but also increasing the homogeneity of the design space. The surrogate optimization model can be set up with different options for all three main components as well as various combinations of them. Since the performance of the surrogate model depends on the problem, it is impossible to define a single option as the “best option”.

This study focuses on the design and the analysis of numerical experiments by using stochastic surrogate models, namely the Kriging model. The main idea in the surrogate model, also known as meta model, is to build a mathematical model between the input and the output regardless of the physics behind the simulations. Surrogate models coming with deterministic results, such as classical regression model, assume that the relation between the input and the output is straight. Predictable points can be defined as the points which are not included in the Design of Experiment and output of the surrogate model. In the deterministic model, confidence level is not an output of the model, and all predictable points can be predicted within the same confidence level. The Kriging surrogate model, on the other hand, consists of both the deterministic and the stochastic part. The stochastic part, as its name signifies, has a random probability distribution and the predictable point may be analyzed statistically but may not be predicted precisely. The Kriging model does not come with a certain result, but with a confidence level. Theoretical part of the Kriging model is dependent on the study of the French mathematician Georges Matheron [1] and its name is coming from South African geostatistician Danie G. Krige [2].

Sampling, known as Design of Experiment (DoE), is related to choosing data in the design space to help surrogate model to describe the space accurately. Full factorial is the best DoE to model the entire space, but it is not possible to put it into practice with multiple inputs and large design spaces. On the other hand, random sampling results in uneven distributions along the space and it can cause missing region in the optimization procedure. Figure 1.1 demonstrates the differences between random sampling and Latin Hypercube Sampling (LHS) [3]. The importance of DoE stems from the determination of the initial point to represent the design space.

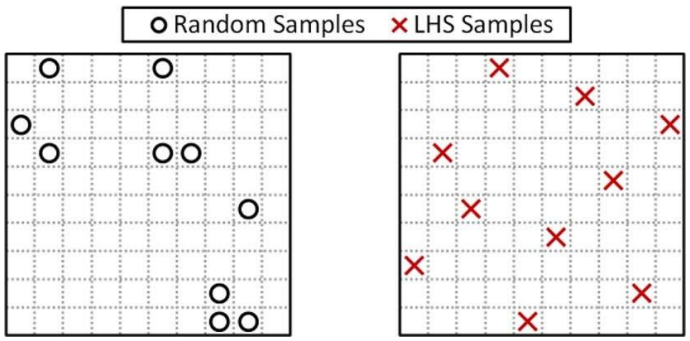


Figure 1.1 Sample distribution for random samples and LHS samples

After sampling, surrogate model is coming into process to fit the behavior of I/O relationship and the data itself. There are various types of surrogate model which can be based on classical deterministic approach such as linear approximation, Response Surface Methodology (RSM), Radial Basis Function (RBF) or on recent stochastic approach such as the Kriging model, Bayesian framework, neural network model. All the aforementioned models are actively used in accordance with the complexity and dimensionality of optimization problem itself. Generally model selection sections in the machine learning literature begin with the sentence of William of Ockham (1287 – 1347) "Entia non sunt multiplicanda praeter necessitate", which can literally be translated as "Entities must not be multiplied beyond necessity", transformed into "Model complexity must not be increased beyond what is necessary to correctly predict the data" for the machine learning society. The Kriging surrogate model consists of deterministic and stochastic parts. The trend in I/O relation is modelled in the deterministic part and generally constant, linear, or low order polynomial functions are used for it. The stochastic part can be modelled as realization of the normal distributed Gaussian stochastic function which has zero mean, spatial covariance and correlation matrix. In the modelling, the main assumption is that the observation points are noise-free points, meaning that the standard deviation in the observations corresponds to zero. There are some approaches about how to include the noise in the calculations [4]. On the other hand, the popularity of surrogate model optimization stems from the success of the design

and the analysis of numerical experiments which is accepted as having noise-free results. Figure 1.2 represents the confidence level and the prediction of a training data for one variable functions.

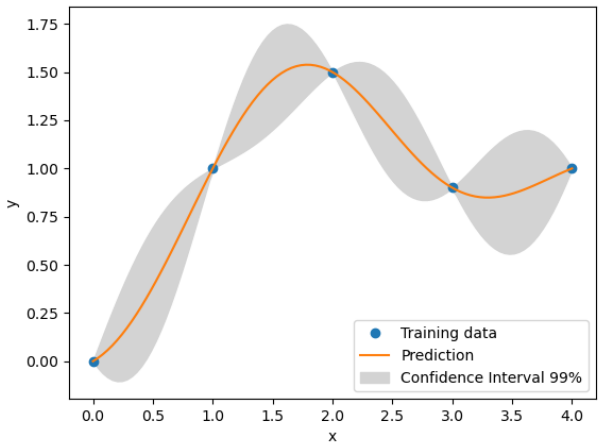


Figure 1.2 Kriging model in 1D space

The Kriging model and the stochastic approach in Bayesian frame extends back a long time. It has been used in many different applications in engineering design cycle; ([5], [6], [7]) for choosing experiments in materials science; medicine molecular design ([8], [9], [10]) and in reinforcement learning ([11], [12]). It became popular thanks to two important studies, namely the study of the Jones et al. [6] and study of Snoek et al. [13]. The study of Jones et al. [6] represents the definition of the acquisition function for selection of the successive point for optimization process which is known as Efficient Global Optimization (EGO), which will be mentioned in the methodology part in detail. EGO proved itself as finding global optima for expensive black-box derivative-free functions [14]. The study of Snoek et al. [13] which is tuning hyperparameters in machine learning, especially very complicated version deep neural networks, supported this idea and made it very popular for the machine learning society and research.

The Kriging surrogate model is suitable to model with few samples compared to neural network which needs more data to fit an accurate model. This situation makes

it suitable for optimization tools with a smaller amount of data. In addition to that, Bayesian inference is well-suited for optimization over continuous domain with less than 20 dimensions and it can handle stochastic noise in function evaluations [9].

1.1 Literature Review

Improvements in the computer hardware which can be summarized as the increase of the CPU power, constructing supercomputer systems with numerous numbers of CPUs and involving GPU in the scientific calculations make the data science and machine learning the most popular topics in the recent years. As a result of this popularity, a vast variation of research on these topics is being conducted all around the world. This study particularly focuses on the main components of the surrogate model optimization which are Design of Experiment (DoE), Kriging surrogate model, Bayesian frame optimization based on engineering problems especially aerodynamics, Gaussian Process Regression (GPR) and acquisition functions.

Aerodynamic design database is based on the analyses and experiments conducted over the decades. On the other hand, unconventional design problems need high-fidelity analyses tools and modern optimization methodologies. There have been several recent thesis studies at METU for aerodynamic design and optimization. The previous thesis study of Dincer [15] [16] was parametric design of grid fins, which are unconventional control surfaces, using Design of Experiment as Box-Behnken, surrogate model as Response Surface Method and CFD as flow solver based on Reynolds-Averaged Navier-Stokes (RANS) equations. Another thesis study was about aerodynamic design and optimization of guided munitions which was conducted by Gun [17] [18]. In this study, DoE was constructed by Sobol algorithms and genetic and simplex algorithms were used to find the optimum point of the design. Oguz's thesis study [19] [20] was also related to an optimization study of horizontal axis wind turbine. Airfoils, parametrized by class-shape transformation and parametric section methods, were optimized with genetic algorithm. Power performance was obtained by blade element theory and aerodynamic data is

calculated from XFOIL open-source airfoil software, which were faster engineering analyses based on simpler aerodynamic theories and tools compared to CFD.

1.1.1 Design of Experiment (DoE)

Design of Experiment (DoE) is the most critical part of the success of the surrogate model. Starting with the full factorial methods causes loss of feasibility for high dimensional and extensive design space. Latin Hypercube Sampling (LHS) is one of the most frequently applied methods in the design and analysis of computer (computational) experiments. It can be explained as random sampling methods, but avoiding clustering samples which means that not duplicating the samples in the variable's rows or columns. On the other hand, it constructs a non-uniform distribution over the design space which makes it perfect to couple with surrogate model based on stochastic design methods.

The implementation of the LHS in computer experiments has been started by study of McKay et al. [21]. The efficiency of the LHS in the computer experiments was first demonstrated in study of McKay et al. [21], Liefvendahl and Stocki [22] and Husslage et al. [23] with avoiding replications. Optimal Space Filling Design (OSFD) can be explained as the improved version of the LHS, and it is also known as Optimal Latin Hypercube Sampling (OLHS) [24]. OLHS is the result of the combination of the LHS and optimal design by optimizing the integrated mean square error of prediction and entropy. Figure 1.3 shows the differences between OLHS and LHS [25]. As it can be seen from Figure 1.3, OLHS has predefined trend in the design space. LHS is preferred as DoE method in this thesis study, since LHS preserves its randomness.

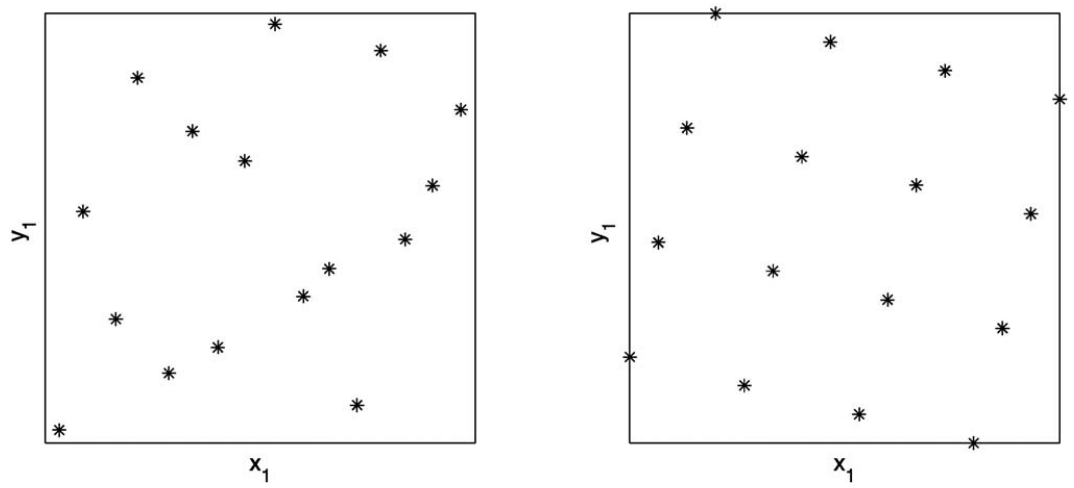


Figure 1.3 LHS samples (right) and OLHS sample (left) distribution

The study of Mukhtar [26] focused on the design optimization in the Computational Fluid Dynamics (CFD) which is conducted by Kriging surrogate model. Results of different DoE methods, Central Composite Design (CCD), Bos-Behnken design (BBD), LHS and OSFD, were compared. According to their results, LHS and OSFD have better results fitting the surrogate models compared to CCD and BBD. DoE methods and surrogate model can be favored as pairs in different applications. BBD comes forward in the polynomial regression model. On the other hand, LHS and OSFD are most favorite in the complex computer experiments with surrogate models such as Kriging surrogate model and Bayesian inference [27].

Determination of the initial points in the design space to build a surrogate model is critical in the optimization procedure. Loeppky et al. [28] showed the required sample size to meet the desired level of accuracy by using Gaussian Process Regression (GPR), which is same with Kriging model. He also tried to answer the validity of the rule of thumbs which is defined as 10 times the number of input variables. One of the examples from this study [28], which was conducted by 4 dimensional LHS with GPR, demonstrated that the improvement coming from increasing the sample size is not dramatic.

1.1.2 Kriging Surrogate Model (Gaussian Process Regression) and Bayesian Optimization

Kriging model, which is also called spatial correlation model, is named after South African geostatistician Krige. Theoretical part was developed by the French mathematician Matheron in 1963 [1]. In this decade, the model was only being applied to 3-dimensions which is coming from the spatial dimension of the geostatistics [29]. After years, the success of the model has led to different application areas such as implementation to the I/O deterministic simulation model. The classical article about design and analysis of computer experiments published by Sacks et al. [30] and then EGO algorithm published by Jones et al. [6] showed that Kriging model is very useful to find optimum points for expensive black box simulation or high-fidelity nonlinear simulations used in engineering.

Simpson et al. [31] showed that Kriging model is very effective in simulation based multidisciplinary design optimization. Classical Response Surface Methods (RSM) and constant Kriging models were investigated to model the real function and to find the optimum points. This study [31] was conducted for aerospike nozzle design, The weight, thrust and gross liftoff weight were modelled by using a metamodel. The results showed that Kriging models have better results concerning R^2 score (R^2) and root mean square error (RMSE).

In Martins and Simpson's study [32], which contains different dimensional test functions´ results, showed that Kriging model has very good results regarding unbiased and Gaussian distributed assumptions in the model. The aim of Martins and Simpson [33] was to figure out the effect of different parameters used in the Kriging models. Model parameter estimation methods, importance of trend function and metamodel error assessment techniques are the parameters which were investigated in that study. The conclusions showed that maximum likelihood estimation (MLE) is the best estimation method, and more complex trend function provides better approximation for deterministic numerical simulation with MLE. The criterion for the best Kriging model was selected by minimizing RMSE.

The Kriging model has been widely used in the different fields of engineering which can be seen in Dellino's studies in different years [34] [35]. They showed that it is very effective to use Kriging model in the control design area [34]. They optimized ionic polymer-metal composites actuators by Kriging metamodel [35].

The detailed study on Bayesian optimization using GPR inherently to fit CFD results was conducted by Morita et al. [36]. Two different cases, which are shape optimization in lid-driven cavity and shape optimization of the wall of a channel flow, were selected to find the optimum design. In the first part [36], the two-dimensional cavity has an open upper part (lid) which has a constant horizontal velocity in the upper boundary and stationary side and lower walls which is called lid-driven cavity as shown in Figure 1.4.

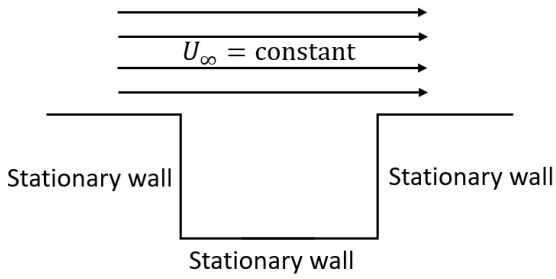


Figure 1.4 Demonstration of lid-driven cavity

In the optimization of lid-driven cavity, the variable was selected as shape of the side wall which was designed to minimize and maximize the energy dissipation due to cavity [36]. In a lid-driven cavity scenario, energy dissipation is a consequence of the fluid's internal friction, turbulence, and the conversion of kinetic energy into heat. The higher the velocity gradient or the viscosity, the greater the shear stress and, consequently, the higher the energy dissipation. The upper and lower boundaries had constant properties; the side walls were optimized. The results of minimum and maximum dissipations taken from the Morita et al. [36] are demonstrated in Figure 1.5 and 1.6.

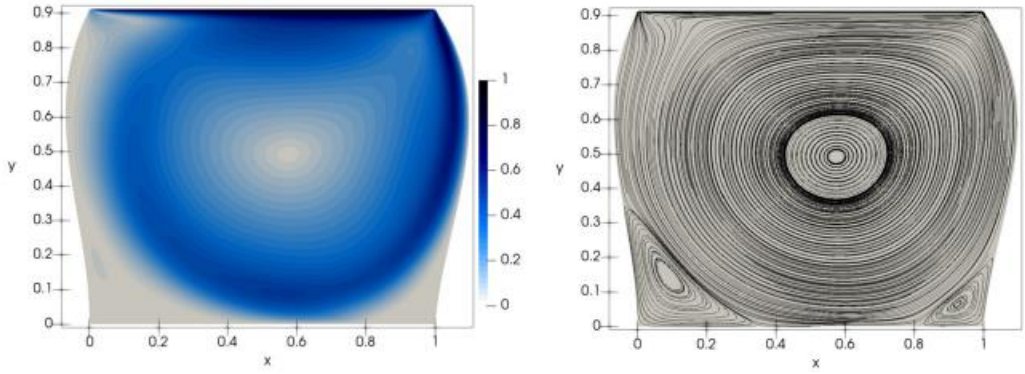


Figure 1.5 Contours of the velocity magnitude (normalized with U_∞) (left) and streamlines of the cavity flow (right) for minimum energy dissipation [36]

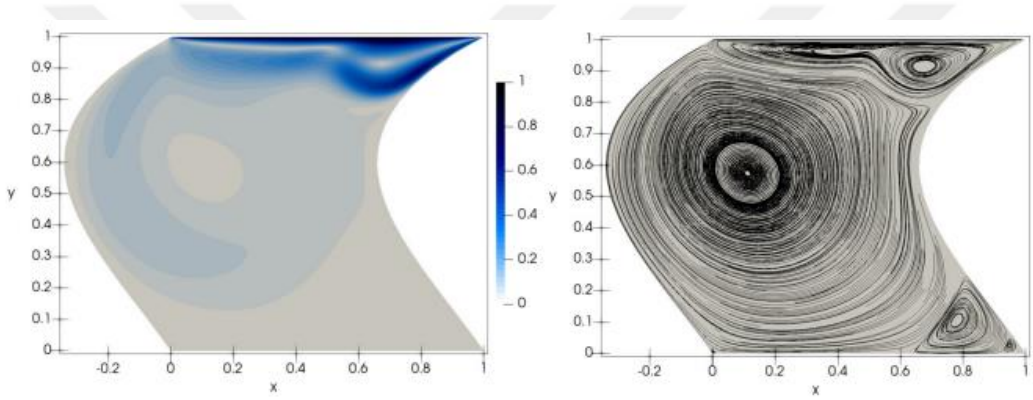


Figure 1.6 Contours of the velocity magnitude (normalized with U_∞) (left) and streamlines of the cavity flow (right) for maximum energy dissipation [36]

In their second case [36], the optimization of diverging channels aiming to optimize the pressure-gradient (PG) turbulent boundary layers (TBL), which is a concept in fluid dynamics, particularly in the study of how fluids flow over surfaces, were studied. Clauser pressure-gradient parameter is commonly used as a non-dimensional parameter related PG TBL. This parameter is related with the displacement thickness, the magnitude of the wall-shear stress, and the pressure gradient along the boundary-layer edge. The objective was to find the desired Clauser PG parameter along the edge of the TBL by optimizing the shape of the upper wall.

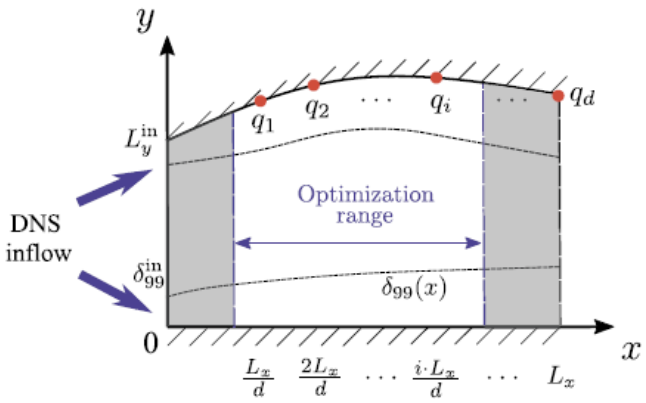


Figure 1.7 Demonstration of the parameters in shape optimization of the diverging channel [36]

First, the scope of the optimization process was to reach a constant Clauser PG parameter (β) in some region of the diverging channel shown Figure 1.8. The design problem was to coincide desired Clauser PG parameter distribution shown as dashed lines with the result of the CFD simulations shown as blue lines in Figure 1.8. Then, the optimization process was repeated to reach a predefined variable Clauser PG parameter (β) distribution in the diverging channel as shown Figure 1.9. The design problem was to coincide the desired distribution shown as dashed lines with the CFD results shown as blue lines in Figure 1.9. For both cases, the results of the optimization process showed that the optimization was done properly.

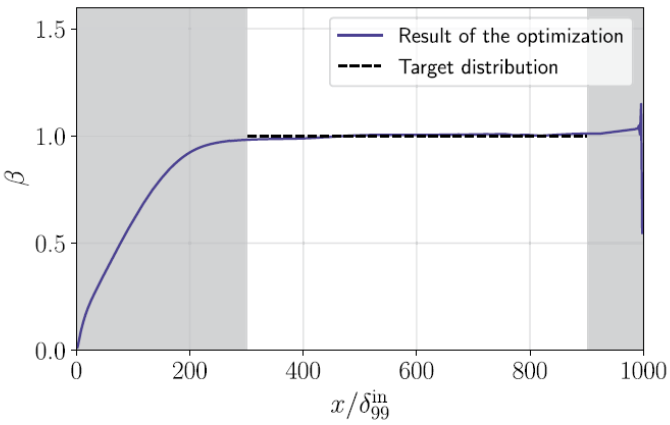


Figure 1.8 Results of the constant Clauser PG parameter optimization [36]

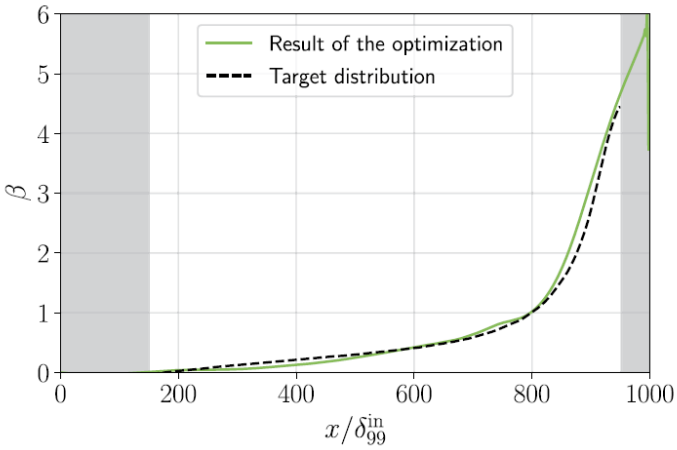


Figure 1.9 Results of the variable Clauser PG parameter optimization [36]

The nature of the Kriging model or GPR, which is a stochastic approach, not only allows uncertain data coming from its nature, but also estimates the uncertainty in their prediction. RSM is also very capable of finding global optimum points, however, it suffers from the curse of dimensionality which needs numerous numbers of samples with increasing number of variables [37]. In the GPR, the decision about the next sample directly takes the predictive uncertainty of the surrogate into account. This active involvement of the uncertainties in the algorithm is another advantage of the Bayesian optimization approach over RSM methods [38]. It is also observed that number of numerical simulations needed to build a reliable surrogate does not change significantly after 8 variables [36].

As computation of gradients using Large Eddy Simulation (LES) needs too much time and computation effort, derivative free optimization techniques are preferred [39]. Surrogate model optimization techniques work quite well for this kind of problem since it filters out the noise in evaluation. In Talnikar et al. study [40], Bayesian optimization was used LES for different cases which are flow control for drag reduction and design of trailing edge of a turbine blade to reduce heat transfer and pressure loss. In the first case, optimization algorithm tries to find the wave speed with the most drag reduction achievement for a constant mass flow rate. It resulted in 60% drag reduction according to the baseline which has no travelling wave. In the

second case, the objective function was defined as a linear combination of the scaled version of the pressure loss and heat loss. B-splines were used to parameterize the trailing edge in two dimensions and piecewise polynomial functions of a specific order were used to represent the TE shape curve. The optimization resulted in 17% reduction in heat loss and 21% reduction in pressure loss.

In Nabae and Fukagata's study [41], Bayesian optimization was used to reduce turbulent friction drag by controlling parameters of travelling wave-like wall deformation. The numerical simulation tool was Direct Numerical Simulation (DNS) with constant pressure gradient conditions. The maximum drag reduction was 60.5% according to the baseline configuration.

In Laurenceau et al. study [42], Reynolds-Averaged Navier-Stokes (RANS) simulations for three-dimensional cases were performed to reach accurate detailed aerodynamic design. It is critical to choose the proper optimization algorithm to reduce the computational cost. RSM is applicable to low dimensional problems, since increasing dimensionality needs high density of samples which makes the optimization process a computationally expensive problem. The first problem was deformation of the upper surface RAE2822 airfoil to reduce the total drag. Six design parameters were determined to deform the upper part of the airfoil to find an optimum shape. The optimized shape resulted in 23.3% drag reduction according to the baseline configuration. In the second problem, 45 shape parameters corresponding to the 15 Hicks-Henne functions on the upper surface of the three-dimensional wing were optimized. The Kriging surrogate model-based optimization algorithm reduced the drag coefficient by 5.4%.

1.2 Aim of the Thesis

This study aims to integrate high-fidelity analytical tools into the design cycle and thereby to reduce the overall duration of the design process through enhanced computational power. An optimization tool based on Bayesian inference is

developed to find the optimum design. In this study, deterministic numerical simulations are employed to construct a Kriging surrogate model, which represents a stochastic relationship between inputs and outputs, independent of the underlying physics of the simulations. Computational fluid dynamics, an expensive high-fidelity tool, is replaced by the Kriging surrogate model in Bayesian optimization. Latin hypercube sampling, which has randomness and homogeneity, is used to initialize the samples in the design space; however, it is not possible to increase the number of samples for iterative processes. Furthermore, this study investigates a hybrid methodology for parallel incremental sampling, which uses randomness and even distribution of Latin hypercube sampling and applicable for any initial quantity of samples to target samples. The optimization tool, which contains hybrid parallel algorithms for sampling, is tested with well-known Rosenbrock function and applied for shape optimization of delta wing using CFD.

1.3 Outline of the Thesis

The thesis study consists of five chapters. In Chapter 1, Kriging surrogate model and Bayesian optimization techniques are introduced regarding their historical background. The motivation and objectives of the thesis and the thesis outline are presented.

In Chapter 2, the literature survey related to the thesis, which are Latin hypercube sampling, Kriging surrogate model and Bayesian optimization, are explained. The application fields with historical background are also mentioned.

In Chapter 3, the methodology of parts of the surrogate model optimization is explained in detail. In addition to that, FLOWUnsteady, which is an unsteady flow solver used in this study is described and its validation case is mentioned in this study. Leave-one-out cross-validation (LOOCV) model used to validate surrogate model and Rosenbrock test function used to test optimization algorithm are also presented.

In Chapter 4, the results of the cases in which surrogate model optimization is applied are presented. Application of optimization starts with finding the minimum points of the test functions which are Rosenbrock functions. In addition to the test function application, the optimization tool is used to find the optimum design point for the delta wing design. Optimization process is investigated for two different cases which are the twist angle optimization case and the multi design parameter optimization case with aspect ratio, taper ratio and sweep angle optimization.

In Chapter 5, critical inferences based on conducted test design studies and recommendations for future works are presented.

CHAPTER 2

METHODOLOGY

Design optimization using surrogate model consists of choosing samples in the design space, building surrogate model instead of using high-fidelity analysis tools and finding the optimum point in the design space using the stochastic approach. Among the sampling methods, Latin Hypercube Sampling (LHS) is used for the determination of initial points in the design space. The Kriging surrogate model, also known as Gaussian Process Regression (GPR), is built to represent the trend of the function which is being tried to be optimized. Optimization cycle using surrogate model is based on the assumption that the model and the real function are having the same trend. Literature proves that this assumption concludes with satisfactory results ([35], [34], [13]). Bayesian inference is used to find the optimum point in the design space.

The design cycle shown in Figure 2.1 starts with the definition of the problem and the limits of the design space. Following that, with the help of Design of Experiment algorithms, which is LHS in this study, initial points are determined. Initial points are analyzed with the high-fidelity analysis tool which is determined as FLOWUnsteady for this design study. Initial points and corresponding results of the solver are used to construct a Kriging surrogate model. Leave-one-out cross-validation (LOOCV) methods are used to analyze the accuracy of the surrogate model and the CFD analysis tool. Well-known performance parameters such as Root Mean Square Error (RMSE) and R^2 score calculated for LOOCV are some of the most reliable validation methods to investigate the accuracy of the model. The next step is to optimize the surrogate model to find minimum/maximum points. Efficient optimization methods for black box functions are gradient-free probabilistic optimization methods and all surrogate models can be accepted as black box functions. Bayesian inference, a statistical inference process, which is based on GPR

and acquisition function, is used to improve/optimize the surrogate model. “Expected Improvement” is selected as an acquisition function. The combination of “Expected Improvement” and Bayesian optimization which is the study of Jones in 1998 is called as Efficient Global Optimization (EGO). The design process should be finalized according to predefined criteria which can be determined as validation of the optimum point, accuracy of the surrogate model and enhancement in the results. The optimum point is then validated with the high-fidelity analysis tool, FLOWUnsteady. If all or some of criteria are not satisfied, the cycle goes back to the sampling step with the addition of new samples to construct again the surrogate model which is called as hybrid incremental sampling method in this study. LHS is a sampling method using randomness regarding the even distribution in the design space, and randomness does not guarantee to use same initial points while updating or increasing the number of the samples with LHS method. Incremental sampling is not an easy task in this process and needs a new approach to achieve it. In this study, a hybrid approach which is a combination of the EGO algorithm and strategy coming from even distribution and randomness of the LHS is developed for incremental sampling. All steps in the design cycle are verified first with a test function, Rosenbrock function, which is used to test the optimization algorithms and strategies. Then, to show the effectiveness of the hybrid optimization algorithm, two different real problems regarding the design of the delta wings are investigated.

The surrogate model optimization algorithms are coded in Python environment. Python is a very popular and powerful coding environment for the machine learning society. Its popularity stems from its efficient libraries which are coded and operated by professional companies and coders. Various libraries about any topic can easily be found and all of them are open sources. Working with open sources gives more control in all steps of the study, in contrast to black box commercial programs.

FLOWUnsteady is a meshless unsteady solver which is coded in Julia program language. Julia has become very popular in recent years, especially with its capabilities in scientific computing, which makes it the top choice for mathematicians.

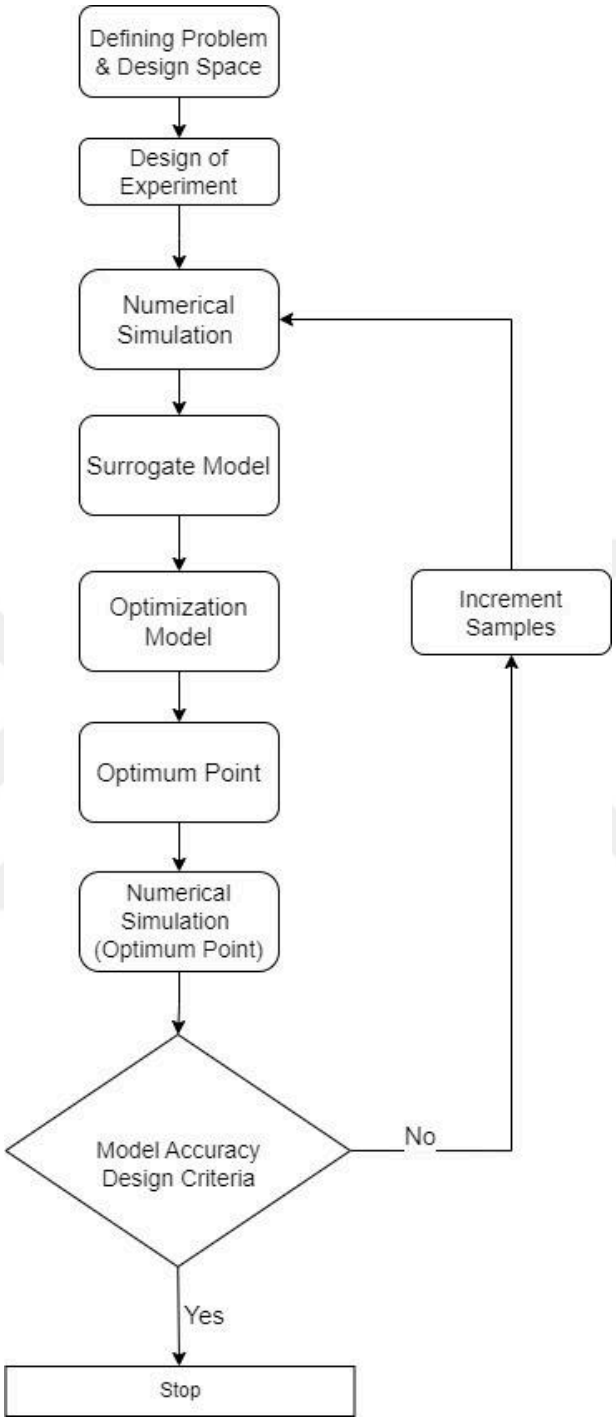


Figure 2.1 Flowchart for surrogate model optimization process

2.1 Rosenbrock Test Function

Before starting an optimization procedure based on expensive analysis tools, various functions can be used for testing optimization algorithms and cycles. Rosenbrock is one of the most popular functions for testing optimization algorithms. It is a non-convex function which can be defined with different number of inputs, and its optimum point is difficult to find because of its banana, or valley shape. $X1$ and $X2$ are input variables of the Rosenbrock functions, and it can be formulized as follows:

$$f(X1, X2) = (1 - X1)^2 + 100(X2 - X1^2)^2 \tag{2.1}$$

Contour plot for 2-dimensional Rosenbrock function is shown in Figure 2.2. The values of the function have a wide range, so the function is preferred to be plotted in the logarithmic scale. The minimum point for Rosenbrock corresponds to (1,1) shown as a red star in the figure with a value of 0.

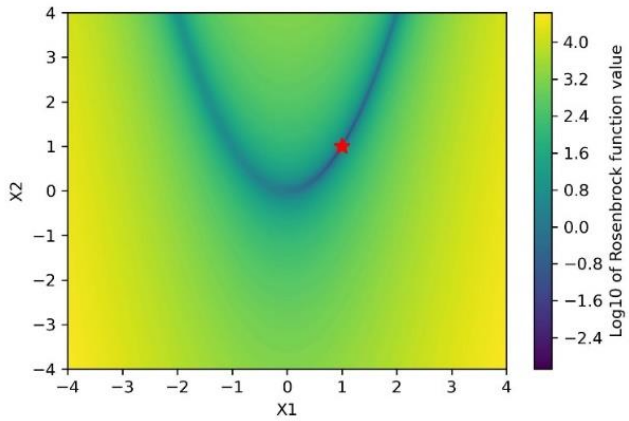


Figure 2.2 Contour of Rosenbrock functions for 2D

2.2 Sampling Algorithms/Strategies

The sampling algorithm is the critical part of the design cycle since it directly affects the required computer power in the design cycle and the accuracy of the surrogate model. In this study, sampling algorithms are used not only for determination of the initial points but also for increasing the number of samples in each iteration. Initial points are determined by the Latin Hypercube Sampling (LHS). Literature shows the effectiveness of LHS in the Kriging surrogate model. In addition to that, optimization tools based on LHS, and surrogate model are suitable for parallelization. On the other hand, no incremental sampling method, which is both effective and can also be parallelized in improving accuracy of the surrogate model, became prominent in the literature. The randomness of LHS leads to no common points with the use of LHS for both equal and different number of initial samples, which makes it impossible for LHS to be used in incremental sampling process. This study proposes a hybrid incremental sampling method for increasing samples which uses both randomness and even distribution of LHS and the result of the Bayesian inference process.

2.2.1 Latin Hypercube Sampling (LHS)

Latin Hypercube Sampling (LHS), which is proposed by McKay et al. [21], is a very popular option for the design and optimization of computer experiments. The idea is to fill the space homogenously regarding the equal distribution of the variables. In the sampling methodology, one of the challenging points is to determine the number of points in the first step. In the literature and applications conclude with a rule of thumbs which corresponds that number of initial points is equal to 10 times the number of input variables [43]. The number of samples depends on the problem itself and should be updated in the process according to criteria to meet the accuracy of the model, on the other hand, the rule of thumbs is a good start.

Distribution of the samples for different number of samples on the Rosenbrock test functions is shown in the Figure 2.3. It can be also realized from Figure 2.3 that there

are no common samples for 20 and 30 initial sampling points. If it is needed to increase from 20 samples to 30 samples in the design cycle iteration, using LHS with increased number of samples comes with all new 30 points. Adding 30 new points requires 30 new high-fidelity numerical simulation runs which can be expensive CFD runs as in this study. As a result, using LHS in determining initial samples seems to be an effective choice, on the other hand using it for incremental sampling in design cycle iteration loses its feasibility. This study proposes a new hybrid incremental sampling method which can use the effectiveness of LHS by keeping its initial points in the following iterations.

There are different libraries, which are especially related to surrogate models and optimizations, that contain the LHS method. PyTorch [44], Scikit-Learn [45], Surrogate Model Toolbox [46] are some of them. In this study, Surrogate Model Toolbox, which is a Python library, is used for modeling and optimization with LHS.

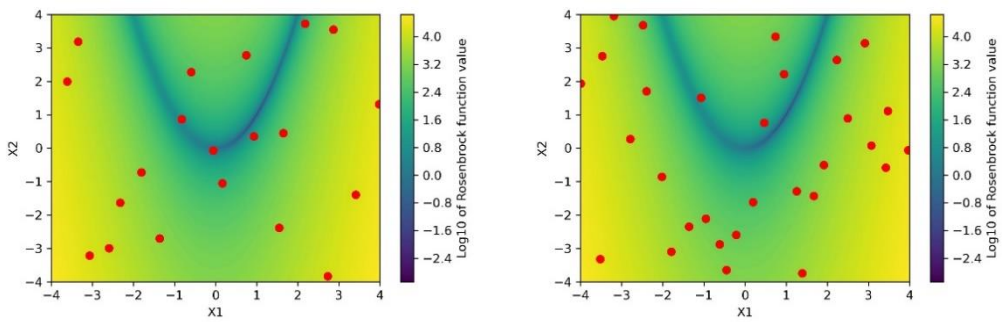


Figure 2.3 Sampling points for the Rosenbrock test function in the design space (left: 20 sampling points, right: 30 sampling points)

2.2.2 Hybrid Incremental Sampling Method

The number and distribution of samples are critical in the surrogate model consistency and accuracy. The potential of LHS stems from both the randomness and the even distribution over the variables in the design space. In the optimization process, it is required to add new samples at each iteration of the design cycle to

improve the consistency and accuracy of the model which results in finding better optimum point. Adding new samples to an initial existing design space to increase the number of samples or creating a new random design space with a new increased number of samples will not be the same. The randomness and even distribution will still be important with increased number of samples. Using LHS will result in a new design space distribution and there may be no common points between the initial and new distribution. This will require expensive numerical CFD simulations to be performed for all the points in the new distribution at each iteration and with an increased number of samples.

The hybrid sampling method, the incremental sampling, which is proposed in this study, will allow the simulation runs to be done only for the new points added to the design space. Initial design space is obtained by LHS, and then the number of samples is increased by using the optimum point from the Bayesian inference and incremental sample method which proposed in this study. Thus, it will combine the effectiveness of LHS with the accuracy of Bayesian inference.

Because of its randomness, adding points is not the same thing as starting with a newer sampling with more points. The new sampling may result in no common points between the initial samples and new increased samples. Having numerical simulations for all initial points makes it impossible/expensive to repeat them. Instead of repeating the simulations for all points, performing simulations only for the newer additional points will be more cost effective.

Using LHS in incremental sampling means doing CFD runs for all new points, and it means numerous numbers of new runs in each iteration which corresponds to very high simulation time. In the incremental sampling method, the required number of CFD simulations is the number of incremental samples. All points are determined in the beginning of the iterations. Numerical CFD simulations for each additional point can be performed in parallel during an iteration of the optimization cycle, thus decreasing the design duration.

It is required to find a better strategy which uses the LHS and adds new samples by keeping the initial points. The hybrid method, which is proposed with this study, consists of two different parts. The first part is to add the optimum point coming from Bayesian inference and the other one is the adaptation of the LHS methodology to incremental samplings.

Incremental samples at each iteration step of the design cycle, which is called delta samples (i.e., δ additional points), is an input variable and depends on the computational power. Delta samples are results of a hybrid algorithm which consists of two different parts and can be parallelized.

The first part comes from the result of the optimization tool as an optimum point which has only one point. In this study, EGO algorithm, which is the combination of Bayesian rules and EI algorithm, is used to find the optimum point. Sample coming from EGO consists of one element and corresponds to the first part. EGO, EI and Bayesian rules are explained in the following sections of the methodology chapter.

It is not possible to use LHS for the delta samples, because the initial sampling and new sampling with more sample points will have no common points by the use of LHS. The second part of the hybrid methodology proposed a new strategy which uses remarkable features of LHS such as even distribution, randomness and parallelization and still keeps the initial points.

For the second part of the incremental sampling methods, the Euclidean norm of each new points obtained using LHS to every initial point are calculated and summed as shown in Figure 2.4. The Euclidean norm in two dimensions means the distance between two points which corresponds the magnitude of the vectors shown in Figure 2.4. After calculating Euclidean norm of each new point to every point in the initial samples, Euclidean norms are summed up for each new point which corresponds to the overall distance of a new point to the initial points. The largest value coming from summation is the outmost or most different point in the updated samples. Thus, the number of required new delta samples (i.e., $(\delta - 1)$ new points), are determined and added as new points into the design space.

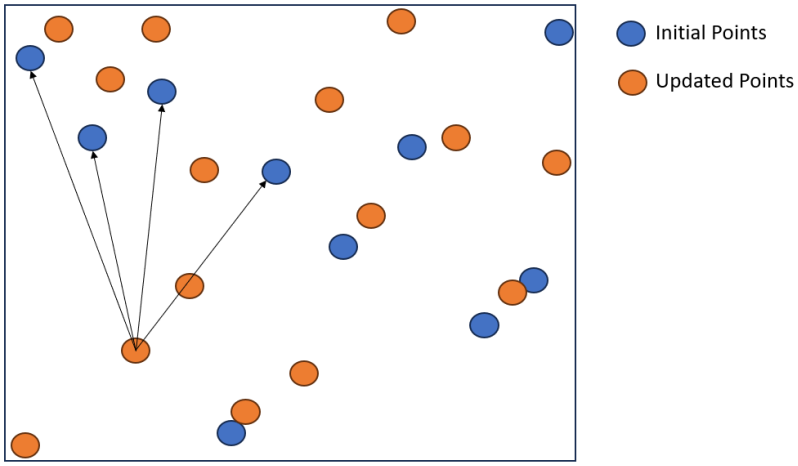


Figure 2.4 Demonstration of difference between the initial points with LHS and the updated points with the incremental sampling method

Formulation for the adding new points are represented as follows:

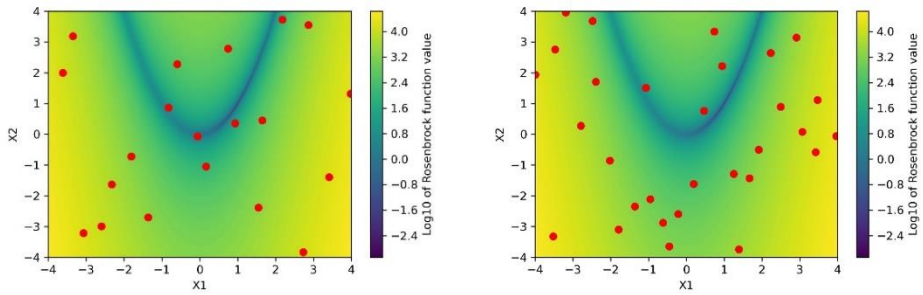
$$distance\ score = \sum_{i=1}^{n} \|x_i - x_u\| \quad \forall x \in A_u \tag{2.2}$$

where, subscript i corresponds to initial point, subscript u corresponds to updated point, n is total number of samples, and A_u is subset of the samples.

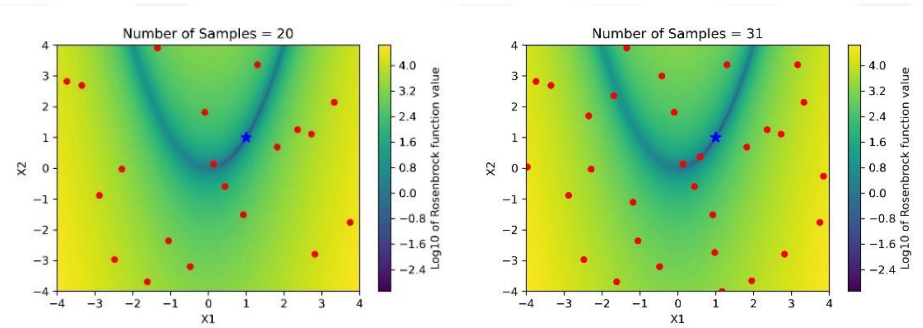
The incremental sampling method can be explained briefly as follows. First, the increase in the number samples is determined and the new points are obtained by LHS method like the initial points obtained by LHS. After that, Euclidean norm of each new point for the new sampling is calculated and then the new points with the highest distance scores are added to the initial Design of Experiment points together with the optimum point obtained with the optimization algorithm.

The new strategy improves finding the global optimum as shown in Figure 3.5. Figure 2.5(a) shows that there are no common points between the initial samples and the new incremental samples. On the other hand, Figure 2.5(b) shows the same initial

points and the new incremental samples that keeps the initial points and additional points obtained by the new strategy that keeps the homogeneity of the design space.



a) Distribution of samples using LHS with 20 and 31 points



b) Distribution of incremental samples with 20 initial and 31 final points

Figure 2.5 Demonstration of difference between LHS and the hybrid incremental sampling method.

2.3 The Kriging Surrogate Model

The Kriging surrogate model is a prediction model between the input and output, it consists of a known function which can be called deterministic part and the realization of stochastic process part. Its popularity is the result of the success in the design based on the deterministic discrete simulations. CFD fits the definition for deterministic discrete simulations. It has the capability of modeling spatial space with a limited set of samples. It is differentiated from other predicting models by making a correlation between samples in spatial space and predict values of the new points by using this correlation coming from spatial space. Basic assumption behind

the Kriging is that the sample points are correlated, and the correlation depends on the distance between the sample points. The distance and direction between the sample points determine the spatial correlation which is used to predict the response surface. In addition to that, Kriging also generates an estimation region for all interpolated regions.

The Kriging surrogate model consists of two main parts which are deterministic part and stochastic part. The deterministic part tries to catch up the general trend of the functions and stochastic part comes with confidence level of the points. The Kriging model is the summation of deterministic part $\left(\sum_{i=1}^{k} \beta_i f_i(x)\right)$ and realization of stochastic process $\left(Z(x)\right)$ and can be formulized as follows:

$$y = \sum_{i=1}^{k} \beta_i f_i(x) + Z(x) \tag{2.3}$$

The deterministic part models the general trend of the function and is kept as simple as it can be. Most common function which is used in deterministic part is constant. Other options for it are linear and low order polynomials. Kriging model with constant deterministic part can be formulated as follows:

$$y = \beta_0 + Z(x) \tag{2.4}$$

Stochastic part assume that points have random distribution in nature and the relations between points are based on the distance and direction between sample points. This idea makes Kriging an effective model to represent different kinds of samples coming from different fields from turning neural network hyperparameters to design of engineering problems. The Stochastic part is a probabilistic model with no mean and only concerning the variance since the trend is coming from the deterministic part. As it is mentioned, the relation of points is based on the distance and direction of the points which concludes with the correlation/covariance matrix containing the relations of each point with the others. As a result, the stochastic

nature can be formulized with the correlation/covariance between the spatial properties with zero mean as follows:

$$cov\left[Z\left(x^{(i)},x^{(j)}\right)\right] = \sigma^2 R\left(x^{(i)},x^{(j)}\right) \tag{2.5}$$

The main assumption here is the points are distributed randomly and the relations are based on the spatial features. Various correlation functions $\left(R\left(x^{(i)},x^{(j)}\right)\right)$ are used to represent the relations between the points. There is no restriction to represent the correlation matrix except being a positive semi-definite matrix. In the recent years, different functions are used as correlation functions and most commons functions are exponential correlation function, squared exponential (Gaussian) correlation function, Matern 5/2 correlation function, and Matern 3/2 correlation function. Gaussian and Matern functions are the most popular correlation functions. In this study, Matern 3/2 correlation function is used.

Common correlation functions are given as follows:

- Exponential correlation function (Orhstein-Uhlenbeck process):

$$\prod_{l=1}^{nx} \exp\left(-\theta_l \left|x_l^{(i)} - x_l^{(j)}\right|\right) \quad \forall\theta_l \in R^+ \tag{2.6}$$

- Squared exponential (Gaussian) correlation function:

$$\prod_{l=1}^{nx} \exp\left(-\theta_l \left(x_l^{(i)} - x_l^{(j)}\right)^2\right) \quad \forall\theta_l \in R^+ \tag{2.7}$$

- Matern 5/2 correlation function:

$$\prod_{l=1}^{nx}\left(1+\sqrt{5}\theta_l\left|x_l^{(i)}-x_l^{(j)}\right|\right.$$
$$\left.+\frac{5}{3}\theta_l^2\left(x_l^{(i)}-x_l^{(j)}\right)^2\right)\exp\left(-\sqrt{5}\theta_l\left|x_l^{(i)}-x_l^{(j)}\right|\right) \tag{2.8}$$
$$\forall\theta_l\in R^+$$

- Matern 3/2 correlation function:

$$\prod_{l=1}^{nx}\left(1+\sqrt{3}\theta_l\left|x_l^{(i)}-x_l^{(j)}\right|\right)\exp\left(-\sqrt{3}\theta_l\left|x_l^{(i)}-x_l^{(j)}\right|\right) \tag{2.9}$$
$$\forall\theta_l\in R^+$$

The Kriging model assumes that the results of samples coming from numerical simulation are accurate which corresponds to no error at samples. This idea makes it popular in numerical experiment studies since most of numerical simulation has repeatability. Unsampled region can be defined as points which has no numerical results and results in this region coming from the surrogate model. Kriging model not only predicts the value of the unsampled region but also finds a confidence interval for it. The predicted function, confidence level and training data for one dimensional function are shown in Figure 2.6. It can be easily seen from the figure that unsampled regions have a wider region which can be decreased by adding points in accordance with the requirements. On the other hand, Figure 2.6 also shows that confidence level for observation points corresponds to a value not an interval.

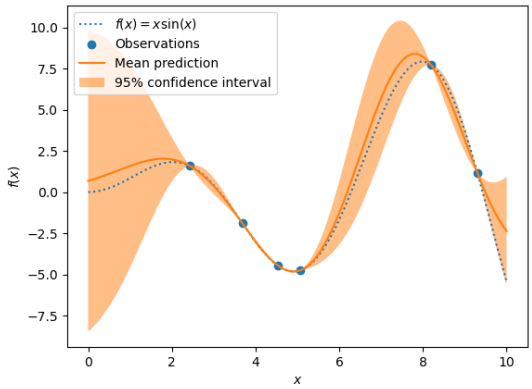


Figure 2.6 Kriging surrogate model for 1D

Kriging surrogate model, also known as Gaussian process regression, is modelled in many well-known Python libraries such as PyTorch [44], Scikit-Learn [45], and Surrogate Model Toolbox [46]. All libraries are coded with the same theoretical background, they have some differences in the interface to control the parameters. Surrogate Model Toolbox is specialized in the surrogate model, especially in the Kriging. It is an open-source library that consists of libraries of surrogate models such as radial basis functions, Kriging, sampling methods and benchmarking problems. The library is jointly coded by the French Government, ONERA, NASA, ISAE, Polytechnic University of Montreal and University of Michigan.

2.4 Leave-One-Out Cross-Validation (LOOCV)

K-fold cross validation is the most common method used in the validation of machine learning models. The best way to measure the performance of the model is to investigate the performance parameters such as root mean square error (RMSE) and R^2 score (R^2) for the non-training data. For this purpose, entire data should be splitted into training and test data. It is obvious that train data should not be used for measuring performance and test data should not be used for training. Although there are some suggestions for splitting in the literature, results highly depend on the discretization and distribution of the test and training data. K-fold cross validation can avoid this dependency by splitting data into k subset and making k times cross

validation for all subsets as shown in Figure 2.7. Each data set is both tested and trained in different models. As a result, this methodology avoids the dependency of the results on chance. K-fold cross-validation concludes with very good results about the coincidence of the model. However, on the other hand, it is not efficient for total times of validation since the same procedure is repeated k times. Leave-one-out cross-validation (LOOCV) is a K-fold cross validation method in which k is exactly equal to the number of data points. In LOOCV, all points except one point are used to train the model and separated one point is used to test the performance of the model. LOOCV is the edge of cross validation which means that cross validation applied as many times as the amount of data. Regarding process time it is totally inefficient for larger data sets such as thousands of samples. In contrast to that, it is very reliable to use in small data sets such as surrogate model optimization for less than 150 samples. For large data sets k is generally taken as 10 and range is changed from 20 to 50 according to computer power and complexity of the model.

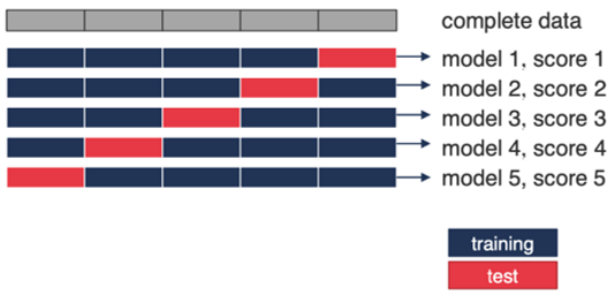


Figure 2.7 K-fold cross-validation

R^2 and RMSE are calculated to investigate the consistency of the model as performance parameters. R^2 is given as follows:

$$R^2 = 1 - \frac{SS_{residual}}{SS_{total}} = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)}{\sum_{i=1}^{n}(y_i - \bar{y}_i)} \tag{2.10}$$

where, n is the number of samples, y_i is the real value of Rosenbrook function, $\widehat{y_i}$ is the predicted value coming from Kriging surrogate model, and $\overline{y_i}$ is the mean of the real value of function or numerical simulation.

RMSE is given as follows:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{n}(y_i - \widehat{y_i})^2} \tag{2.11}$$

where, n is the number of samples, y_i is the real value of function or numerical simulation, and $\widehat{y_i}$ is the predicted value coming from Kriging surrogate model.

2.5 Bayesian Optimization

Bayesian optimization, first defined by Marcus and Blumenthal [47], is a derivative free optimization technique which has an ability to find the global minima/maxima point for very expensive black box functions. It became very popular with the achievement of the tuning hyperparameter in neural networks [13]. Common points for application of the Bayesian optimization are optimizing functions which are difficult to calculate. Bayesian optimization does not care about the physical background and acts to any functions as a black box. In the optimization problems, black box generally uses for the function which is expensive to calculate the derivatives of the functions and itself.

The strategy underlying the Bayesian process is to act as a random function and make a prior relation on the function. It is assumed that prior relation captures the behavior of the function. After evaluating the functions over the sample points, prior is updated to posterior distributions, which concludes with an acquisition function to find the extreme points in the space. Bayesian optimizations can be grouped into sampling, Gaussian regression, Bayes rules and acquisition function. Sampling methodology is quite the same as the sampling methods which are discussed in the

first sections of this chapter. Bayes rule can be formulated as follows with variables which are P(A) prior probability, P(A|B) posterior probability, P(B|A) likelihood probability, P(B) marginal probability:

$$P(A|B) = \frac{P(A)\,P(B|A)}{P(B)} \tag{2.12}$$

Bayes rules has capability of the converting likelihood probability, P(B|A), to posterior probability, P(A|B). These two probabilities are totally different and the best way to explain difference of these two probabilities and Bayes rules is to explain with an example based on the Covid test and blood with SARS-CoV-2 virus. As an instance, the probability of a patient has positive Covid test result (B) given that they have the disease (A) is P(B|A) and the probability that a patient has the disease (A) given that the patient has the positive test result (B). In this example, P(B|A) is the result of the clinical research, on the other hand any patient with the probability of disease with positive test result is P(A|B). Bayes rule has a capability of finding relation between two different but highly related probabilities.

2.5.1 Gaussian Process Regression (GPR)

Gaussian regression, which is same with the Kriging model, is a stochastic approach behind the Bayesian optimization. The methodology of the Kriging model is explained in the previous section. In this section, application of GPR on the samples are explained. GPR assumes that any finite number of samples have no certain relation to each other. In the Bayesian process, prior distribution is acting like a random distribution in nature. GPR uses random distribution as a prior distribution to make a probabilistic model with a specific mean vector and a covariance matrix.

The mean vector can be defined by evaluating the mean function at every point. As mentioned before, GPR and Kriging model assumes that points have random distribution in space. The relation between points is based on the distance and

direction of the points to each other. As a result, constructing covariance matrix is calculated by covariance functions or kernels for each pair of the points which is based on the relations between the points. Kernel functions Σ_0 and covariance functions have the same terminology and the aim of both are to find a relation between points in the sample space and construct a covariance matrix which consists of relations of the point pairs. As expected, the diagonal elements of the covariance matrix is 1 which corresponds to the relation of the point to itself. For the most distant point pairs would have nearly zero covariance value. The kernel functions, which have large positive correlations between close points, are generally used. Only requirement for the correlation matrix is to be a positive semi definite. The idea of the correlation depending on the distance in the spatial space, is the same with the Kriging model. Prior distribution, which assumes random distribution, for observed k points can be related with zero mean and kernels as follows:

- $x_{1:k}$ is the finite collection of points: $x_1, .., x_k$
- $f(x_{1:k})$ is the function value at these points: $[f(x_1), \dots, f(x_k)]$
- $\mu_0(x_{1:k})$ is the mean function at these points: $[\mu_0(x_1), \dots, \mu_0(x_k)]$
- $\Sigma_0(x_{1:k}, x_{1:k})$ is the covariance matrix:
  $[\Sigma_0(x_1, x_1), \dots, \Sigma_0(x_1, x_k); \Sigma_0(x_1, x_1), \dots, \Sigma_0(x_1, x_k)]$
- $f(x_{1:k})$ is the normal distribution: $Normal\big(\mu_0(x_{1:k}), \Sigma_0(x_{1:k}, x_{1:k})\big)$

As it is mentioned, stochastic model building by a given data corresponds to prior probability. The conditional probability or posterior probability for any point can be inferred by using the Bayes inference. Main idea is to infer the new points, which are not in data set and can be evaluated as posterior probability, by using given data set which is evaluated as prior probability.

x_n is any point in the space and tried to be estimated in some confidence level. Confidence level can be investigated by a stochastic model which consists of posterior mean and variance value. The posterior mean of x_n, $\mu_n(x_n)$, is a weighted average between the prior $\mu_0(x_{1:k})$ and an estimation based on the value of the function $f(x_{1:k})$, with a weight that depends on the kernel $\Sigma_0(x_{1:k}, x_{1:k})$ which

depends on distance and direction in space. The posterior variance $\sigma_n^2(x_n)$ is equal to the prior covariance $\Sigma_0(x_{1:k}, x_{1:k})$ less a term that corresponds to the variance removed by observing $f(x_{1:k})$. Relations for functions, mean value and variance are given as follows:

- Normal distribution: $Normal\big(\mu_n(x_n), \sigma_n^2(x_n)\big)$
- Conditional probability: $f(x_n)|f(x_{1:k}) \sim Normal\big(\mu_n(x_n), \sigma_n^2(x_n)\big)$
- Mean:

$$\mu_n(x_n) = \Sigma_0(x_n, x_{1:k})\Sigma_0(x_{1:k}, x_{1:k})^{-1}(f(x_{1:k}) - \mu_0(x_{1:k})\,) + \mu_0(x_n) \tag{2.13}$$

- Variance:

$$\sigma_n^2(x_n) = \Sigma_0(x_n, x_n) - \Sigma_0(x_n, x_{1:k})\Sigma_0(x_{1:k}, x_{1:k})\Sigma_0(x_{1:k}, x_n) \tag{2.14}$$

where, covariance (kernel) function $\Sigma_0(x, x_n)$ and zero mean value $\mu_0(x)$, can be obtained by different approaches.

Different kernels can be used in the calculation of covariance (kernel) function if it satisfies the conditions for covariance matrix which is to be a semi-definite matrix. Commonly used covariance kernel functions, which are same with Kriging model, are are power exponential (Gaussian) kernel and Matern kernel as given in the equations below:

- Gaussian covariance function:

$$\Sigma_0(x, x_n) = \alpha_0 \exp(-\|x - x_n\|^2) \tag{2.15}$$

- Matern covariance function:

$$\Sigma_0(x, x_n) = \alpha_0 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\sqrt{2\nu}\|x - x_n\|\right)^{\nu} K_\nu \exp\left(\sqrt{2\nu}\|x - x_n\|\right) \tag{2.16}$$

For zero mean value function, the most common choice is a constant value. When the function is believed to have a trend, it can also be selected as a low order polynomial which is formulated as below:

$$\mu_0(x) = \mu + \sum_{i=1}^{l} \beta_i \Psi_i(x) \tag{2.17}$$

The parameters in the covariance and mean functions should be defined and there are three main approaches to determine them are maximum likelihood estimate (MLE), maximum a posteriori (MAP) and fully Bayesian approach.

2.5.2 Acquisition Function

Gaussian regression concludes with posterior distribution for any points. Next step is how to determine the next sample using surrogate model and posterior distributions. Acquisition function is defined to find the following points in the space. The most common and successful one is "Expected Improvement (EI)" which is also selected in this study. Its success comes with a special name for this algorithm, EGO which is basically Bayesian optimization using EI. EI is instead of looking for the best improvement point, searching for expected improvement in the next point. The algorithm starts with the finding maximum point in the data set using the following definitions:

- $x_{1:k}$ is the finite collection of points: $x_1, .., x_k$
- $f(x_{1:k})$ is the function value at these points: $[f(x_1), \dots, f(x_k)]$
- $f^*(x_m)$ is the maximum value of function in the data set:

$$f^*(x_m) = \max_{m<k} \mathrm{f}(x_m) \tag{2.18}$$

The new point, x_n, can be anywhere in the space and it can be evaluated as $f(x_n)$. After new evaluation, the best point will either replace the best one $\mathrm{f}(x_n)$ $(if\ f(x_n) \geq f^*(x_m))$ or $f^*(x_m)$ $(if\ f(x_n) \leq f^*(x_m))$. The improvement for the observed value can be expressed as $f(x_n) - f^*(x_m)$ and it can be either positive or 0 and is formulated as:

$$[f(x_n) - f^*(x_m)]^+ \ where\ a^+ = \max(a, 0) \tag{2.19}$$

Expected improvement, where E_n, and I means "expected" and "improvement", respectively, is given as follows:

$$EI_n(x_n) = EI_n\big(x_n | x_{1:k}, f(x_{1:k})\big) \quad \text{with posterior distribution} \tag{2.20}$$

$$EI_n(x_n) := E_n[[f(x_n) - f^*(x_m)]^+] \tag{2.21}$$

The expected improvement can be expressed in closed form by integration by part as:

$$EI_n(x_n) = [\Delta_n(x_n)]^+ + \sigma_n(x_n)\varphi\left(\frac{\Delta_n(x_n)}{\sigma_n(x_n)}\right) - |\Delta_n(x_n)|\Phi\left(\frac{\Delta_n(x_n)}{\sigma_n(x_n)}\right) \tag{2.22}$$

$$\Delta_n(x_n) := \mu_n(x_n) - f^*(x_m) \tag{2.23}$$

where, $\varphi(\cdot)$ is the cumulative density function of standard normal distribution and $\Phi(\cdot)$ is the probability density function of standard normal distribution.

Expected improvement algorithm finds the optimum points as:

$$x_{n+1} = argmax\big(EI_n(x_n)\big) \tag{2.24}$$

It is inexpensive to evaluate or easy to evaluate the 1st and 2nd derivatives of the expected improvement function. The 1st and 2nd derivatives of the expected improvement (EI) acquisition function can be computed using numerical methods. Monte Carlo integration and important sampling are the two main numerical methods in numerical calculation of EI. This can be useful for optimizing the acquisition function in order to improve the performance of Bayesian optimization. Theoretical background is explained detailed in [4]. Other well-known acquisition functions are entropy search and knowledge gradient.

2.6 Unsteady Flow Solver (FLOWUnsteady)

FLOWUnsteady, developed in the doctoral study of Avarez [48], is an open-source unsteady flow solver for unsteady aerodynamics and aeroacoustics which is coded in Julia. Julia is a high-level, general-purpose dynamic programming language, most commonly used for numerical analysis and computational science. The theory behind the solver is the reformulated Vortex Particle Methods (rVPM) which is a mesh-free CFD solver for Large Eddy Simulation (LES), solving the filtered incompressible Navier-Stokes equations in their vorticities form. The curl of the Navier-Stokes linear momentum equations results in the vorticity equation for an incompressible fluid which is given as:

$$\frac{D}{Dt}\omega = (\omega \cdot \nabla)u + v\nabla^2\omega \tag{2.25}$$

where, D is total derivative, $u(x,t)$ is the velocity field, v is kinematic viscosity, and $\omega(x,t) = \nabla \times u(x,t)$ corresponds to vorticity fields. This equation only depends on ω since u can be found from ω by Biot-Savart law. Derivation of the LES version

of the vorticity equations is obtained from tensor notation of vorticity equation and given as follows:

$$\overline{\frac{\partial \omega_i}{\partial t}} + \overline{u_j \frac{\partial \omega_i}{\partial x_j}} = \overline{\omega_j \frac{\partial u_i}{\partial x_j}} + v\overline{\nabla^2 \omega} \tag{2.26}$$

Nonlinear terms in the equations given as $\overline{u_j(\partial \omega_i / \partial x_j)}$ and $\overline{\omega_j(\partial u_i / \partial x_j)}$ cannot be calculated directly, but can be approximated through a tensor T_{ij} which can be evaluated as given below:

$$\overline{u_i \omega_j} = \overline{u_i}\,\overline{\omega_j} + T_{ij} \tag{2.27}$$

The gradient of T_{ij} and its transpose are given as:

$$\begin{aligned} \frac{\partial T_{ij}}{\partial x_j} &= \overline{\omega_j \frac{\partial u_i}{\partial x_j}} - \overline{\omega_j} \frac{\partial \overline{u_i}}{\partial x_j} \\ \frac{\partial T'_{ij}}{\partial x_j} &= \overline{u_j \frac{\partial \omega_i}{\partial x_j}} - \overline{u_j} \frac{\partial \overline{\omega_i}}{\partial x_j} \end{aligned} \tag{2.28}$$

After substituting these equations into the vorticity equation, it becomes as below:

$$\frac{\partial \overline{\omega_i}}{\partial t} + \overline{u_j} \frac{\partial \overline{\omega_i}}{\partial x_j} = \overline{\omega_j} \frac{\partial \overline{u_i}}{\partial x_j} + v\nabla^2 \overline{\omega} - \frac{\partial T'_{ij}}{\partial x_j} + \frac{\partial T_{ij}}{\partial x_j} \tag{2.29}$$

where, the term $\partial T'_{ij}/\partial x_j$ represents the SFS contributions arising from the advective term, whereas $\partial T_{ij}/\partial x_j$ represents the contributions arising from vortex stretching.

The accuracy of LES hinges on the modelling of this tensor. $(E_{adv})_i = \partial T'_{ij}/\partial x_j$ is the SFS vorticity advection; $(E_{str})_i = -(\partial T_{ij}/\partial x_j)$ is the SFS vortex stretching and the equation becomes:

$$\frac{d}{dt}\overline{\omega} = (\overline{\omega} \cdot \nabla)\overline{u} + v\nabla^2\overline{\omega} - E_{adv} - E_{str} \tag{2.30}$$

The PDE can be splitted into inviscid and viscous parts as:

$$\frac{d}{dt}\overline{\omega} = \left(\frac{d}{dt}\overline{\omega}\right)_{inviscid} + \left(\frac{d}{dt}\overline{\omega}\right)_{viscous} \tag{2.31}$$

where,

$$\left(\frac{d}{dt}\overline{\omega}\right)_{inviscid} = (\overline{\omega} \cdot \nabla)\overline{u} - E_{adv} - E_{str} \tag{2.32}$$

$$\left(\frac{d}{dt}\overline{\omega}\right)_{viscous} = v\nabla^2\overline{\omega} \tag{2.33}$$

In the Alvarez study [48], the viscous diffusion term is solved by the core spreading method which is coupled with the radial basis function interpolation approach developed by Barba [49].

The solver uses the Lagragian (meshless) scheme which can not only prevent the disadvantages coming from generation of mesh, but also retain the structure of the vorticity over the distance while minimizing the numerical dissipation. The discretization of the vorticity equation with Lagragian elements is called as representation with vortex particles where each particle represents the volume of fluid that is convected by local velocity field carrying an integral quantity of vorticity. Γ_p is the vortex strength, x_p is the position of vortex particle and δ is the Dirac delta. The vorticity field can be approximated as follows:

$$\omega(x,t) \approx \sum_p \Gamma_p(t)\delta\left(x - x_p(t)\right) \tag{2.34}$$

Finally, the governing equation for the vortex strength [48] is given as follows:

$$\frac{d\Gamma_{\mathrm{p}}}{dt} = (\Gamma_{\mathrm{p}} \cdot \nabla)u(x_p) + 3\Gamma_{\mathrm{p}}\frac{1}{\sigma_p}\frac{\partial \sigma_p}{\partial t} + \frac{1}{\zeta_{\sigma_{\mathrm{p}}}(0)}\left(-M_p^o + M_p^1 + M_p^2\right) - \frac{1}{\zeta_{\sigma_{\mathrm{p}}}(0)}\left(E_{adv}(x_p) + E_{str}(x_p)\right) \tag{2.35}$$

The M terms in the equation represent the effect of the neighboring particles to each other's. The main disadvantage of classical VPM is the numerical instability which causes a breaking structure of the vorticity in the turbulent region. The flow regions in the wake of a propeller are shown in Figure 2.8.

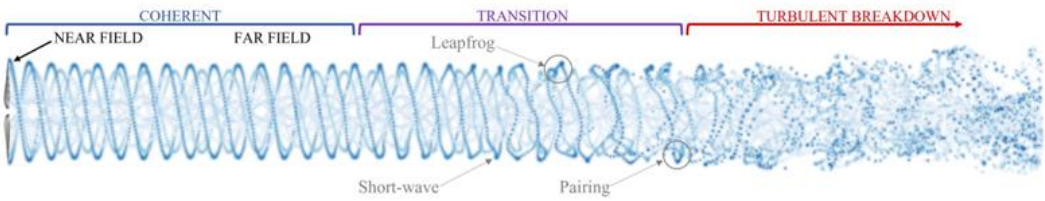


Figure 2.8 Flow regime in the wake of the propeller [48]

Generally, classical VPM formulation is ignored the filtered terms which shows up E_{adv} and E_{str} in the formulation. In addition to that, M terms, which are neighbor effect of the close particles, are also vanished in the solution of the classical VPM solution. Classical VPM also assumes that the shape of particle is constant along the streamlines which means that $\partial\sigma_p/\partial t = 0$. This assumption violates the conservation of mass and momentum in the transition and turbulent regimes which is accepted as the main reason of the numerical instability in the classical VPM.

FLOWUnsteady avoids this disadvantage by reformulating VPM as LES which is numerically stable which also does not increase the computational time. The advantages of the mesh-free solver are listed as not suffering from dissipation due to mesh itself, integration with coarser discretization without losing physical accuracy,

and derivatives being calculated analytically. Most advantageous feature is that VPM solver is 100 times faster than conventional CFD solvers using mesh. FLOWUnsteady has capability of solving tilting wings and rotors, and rotors with variable RPM and variable pitch. It can also be used for different levels of fidelity which is demonstrated in Figure 2.9.

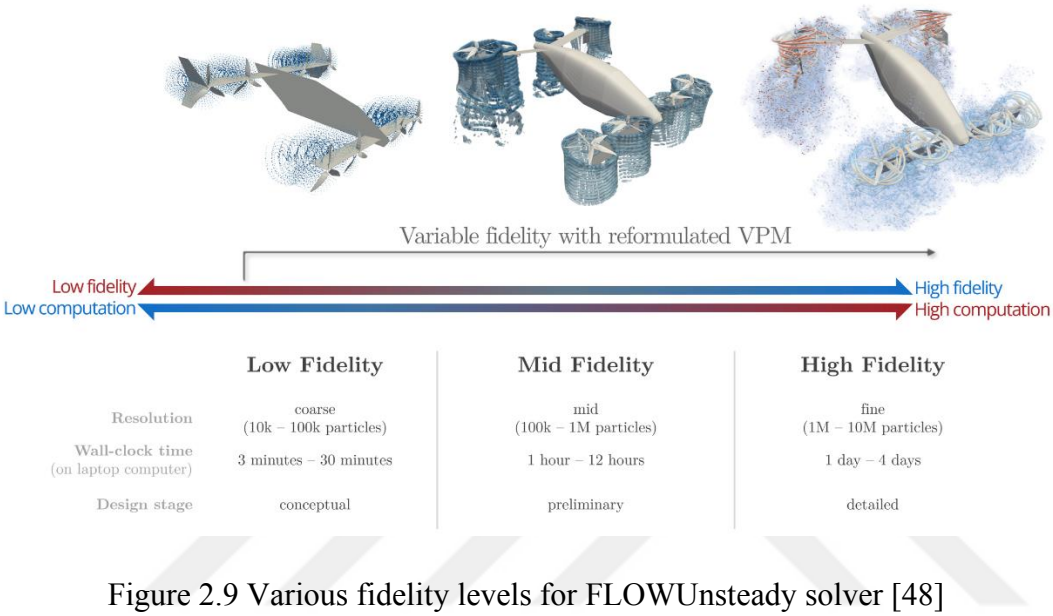


Figure 2.9 Various fidelity levels for FLOWUnsteady solver [48]

Governing equation of vortex strength is given in Equation (2.35) above. Classical VPM assumes that $\partial\sigma_p/\partial t = 0$, in contrast to that, reformulating VPM formulates the change of the particle shape by conserving angular momentum of the particles. Deformation on the particle intensifies the vorticity in the direction that the element is stretched as shown in Figure 2.10.

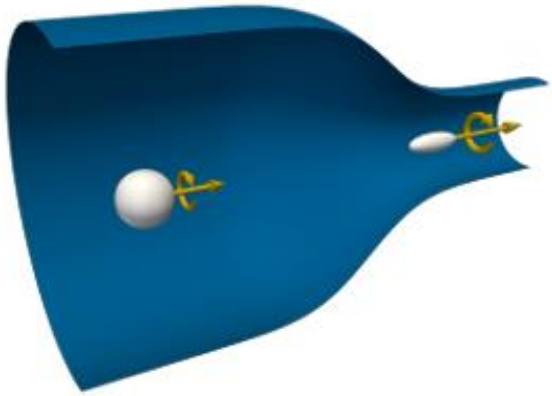


Figure 2.10 Stretching of a spherical fluid element [48]

The other approach for modeling deformation of fluid particles is conservation of mass by using existence of vortex lines and vortex tubes. In this approach, vorticity evolves as the material lines, being identically stretched, and reoriented by the velocity field. The vortex lines are represented as the existence of lines of vorticity and the vortex tube is represented as the surface formed by all the vortex lines forming a closed curve as shown Figure 2.11.

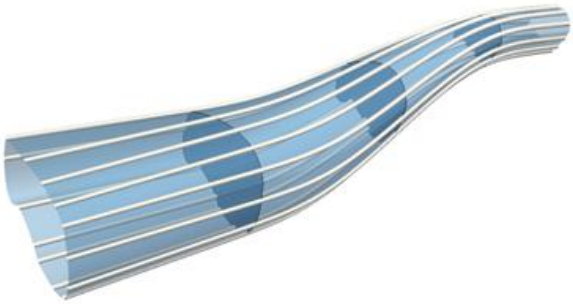


Figure 2.11 Vortex tube formed by vortex lines [48]

For conservation of angular momentum and mass, by using vortex lines and vortex tubes and the governing equations for the deformation of particles and the change of vortex strength are given as below:

$$\frac{\partial \sigma_p}{\partial t} = -\frac{1}{5}\frac{\sigma_p}{\|\Gamma_\mathrm{p}\|}\left[(\Gamma_\mathrm{p} \cdot \nabla)u(x_p)\right] \cdot \hat{\Gamma}_\mathrm{p} \tag{2.36}$$

$$\frac{d\Gamma_{\mathrm{p}}}{dt} = (\Gamma_{\mathrm{p}} \cdot \nabla)u(x_p) - \frac{3}{5}\{[(\Gamma_{\mathrm{p}} \cdot \nabla)u(x_p)] \cdot \widehat{\Gamma_{\mathrm{p}}}\}\widehat{\Gamma_{\mathrm{p}}} \tag{2.37}$$

2.6.1 Validation of Unsteady Flow Solver (FLOWUnsteady)

The validation test case of unsteady flow solver for delta wing is investigated in the Avarez's doctoral study [48] for FLOWUnsteady. The experiment, used in the validation of the low-speed flow over the 45-deg swept back wing, was done by Weber et al. [50]. Explanations of the geometric parameters of the delta wing used in the wind tunnel experiments are shown in Figure 2.12.

Flow conditions in the wind tunnel are tabulated in Table 2.1. The comparison of the experiment and the current numerical solution of FLOWUnsteady are shown in Figure 2.13 and tabulated in Table 2.2.

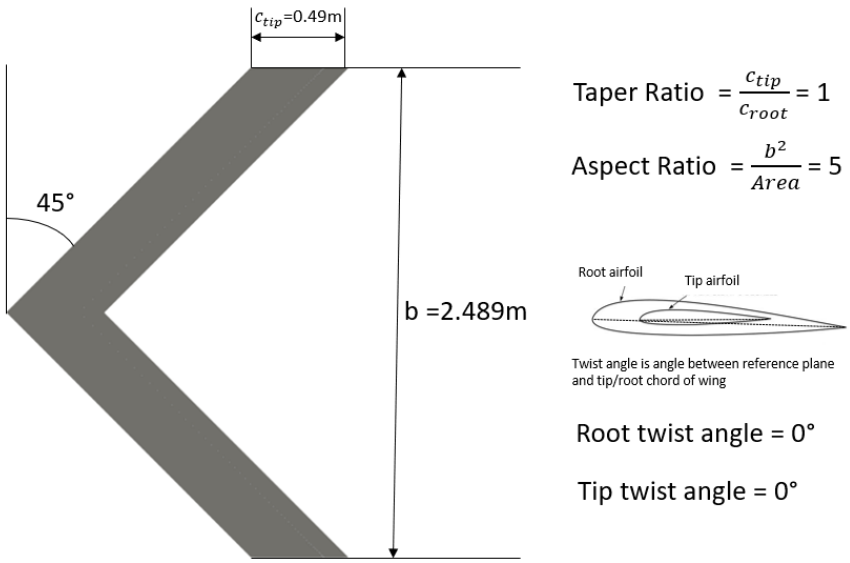


Figure 2.12 Geometric parameters of the baseline wing and explanations of the parameters

Table 2.1 Wind tunnel flow condition

Test Condition	*Values*
Free stream velocity (V_∞)	49.7m/s
Reynolds Number (Re)	1.05million
Angle of Attack (α)	4.2°

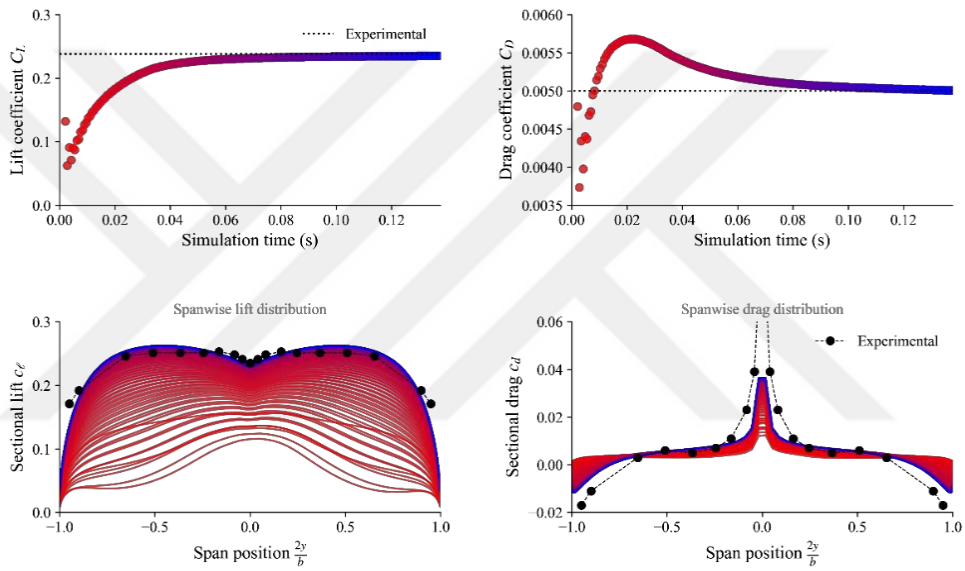


Figure 2.13 Comparison of numerical simulation and experimental results [48]

Table 2.2 Comparison of numerical simulation and experimental results

Test Condition	C_L	C_D	C_L/C_D
Numerical Solution Result	0.238	0.005	48.57
Experimental Result	0.2351	0.00498	47.02
Percentage Error	1.2%	0.4%	3.2%

It can easily be seen from both Table 2.2 and Figure 2.13, FLOWUnsteady has consistent results with the experiment. As a result, it has accurate results for use in the design process of the delta wing. The required computational time duration for one run is approximately 5 min with Apple MacBook Air with M1 chips.

2.7 Optimization Tool Development

The optimization tool, which is called as IncSampMeth_BayesOpt and used in this study, is developed by an open-source programming language, Python. Python is a high-level, general-purpose programming language which supports structured, object-oriented and functional programming. Guido van Rossum started working on Python in 1980s and the 1st release version is in 1991. Python's popularity increases with machine learning and machine learning society takes it as primary programming language. Consequences of it, Python has numerous numbers of libraries for the same topic from different coders, societies, or companies. As a result, Python is preferred for development of the optimization tool in this study.

Similar to other popular programming languages, Python is also object-oriented. The optimization tool is organized by class which makes the code easy to improve and comprehensible. The classes in the program which are demonstrated in Figure 2.14 represent the different parts of the optimization cycle. The main file calls the class files which are DoE, numerical simulation, surrogate model, optimization and visualization. The DoE class is responsible for the initial sampling method which is LHS and the hybrid incremental sampling method which is developed in this study. The numerical simulation class consists of organizing the folders, modification of the input files regarding DoE for CFD runs, calling Julia and running FLOWUnsteady solver, and reading results from the output files of FLOWUnsteady. The surrogate model class constructs the Kriging model, validates with the cross-validation method, LOOCV, and calculates the performance parameters of RMSE and R^2. The optimization class is responsible for the Bayesian inference and EGO

algorithm. The optimization tool code is available on GitHub, https://github.com/emresunay/IncSampMeth_BayesOpt.

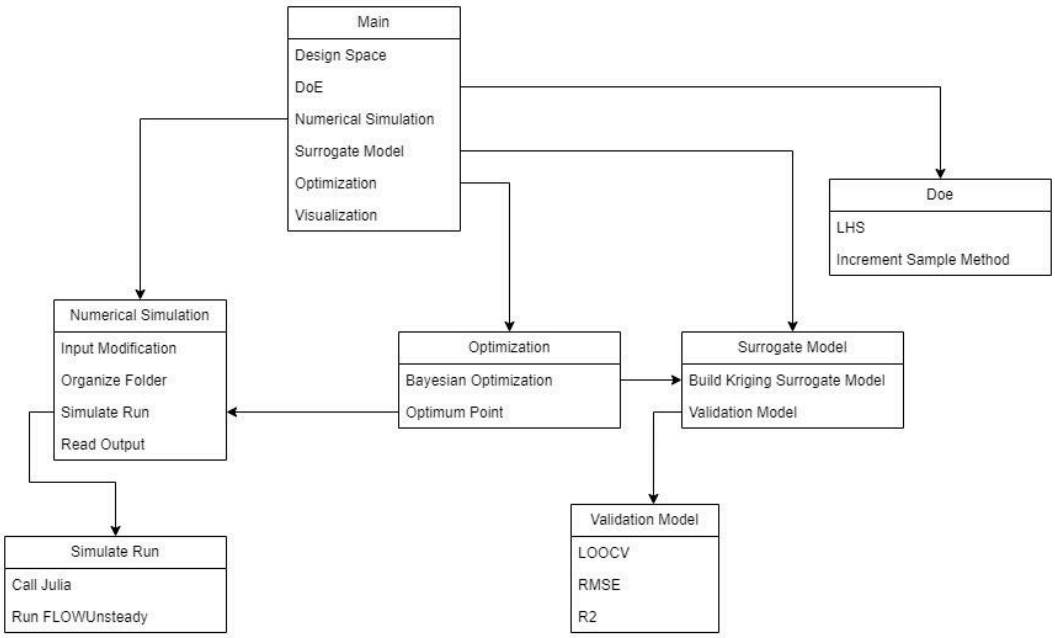


Figure 2.14 Class scheme of the optimization tool: IncSampMeth_BayesOpt

CHAPTER 3

RESULTS AND DISCUSSION

The aim of this study is to develop an optimization tool which is based on stochastic approach and can be parallelized and updated if needed. Previous sections include the introduction about the Kriging model and Bayesian inference, literature survey of each component of the optimization tools and theoratical background behind the optimization tool and numerical analysis tool, FLOWUnsteady.

In this section, one test function and two different design problems are investigated to present the performance of the optimization tool.

The first case is to find the minimum point of the Rosenbrock function which is used to test the performance of the optimization tool. It has a valley or banana shape, as shown in Figure 3.1, which causes difficulty in finding the optimum point. Classical optimization methods are struggling to find the optimum point of the Rosenbrock function. In the first part of the results, surrogate model optimization is tested to find the optimum point of the Rosenbrock functions.

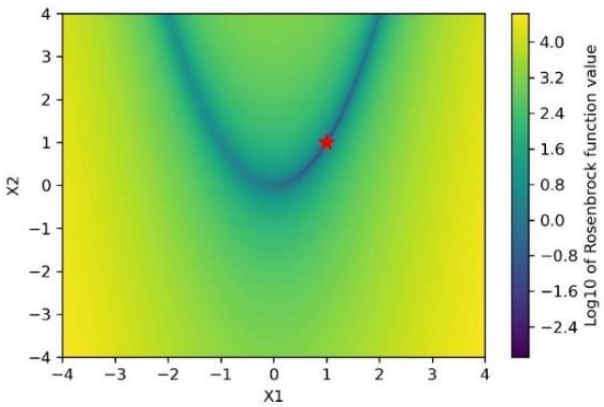


Figure 3.1 Contour of Rosenbrock functions and optimum point

In the second part of the results, surrogate model optimization tool is applied for practical relevance which is determined as shape optimization of delta wing using CFD. Two different design problems, which are optimizing the root and the twist angle so that maximizing C_L/C_D; optimizing aspect ratio, taper ratio and sweep angle so that maximizing C_L/C_D, are investigated. One of the aims of this study is to make high-fidelity tool part of design cycle. FLOWUnsteady is selected as a CFD tool which corresponds to the high-fidelity tool in the design cycle and 45-deg swept back wing experimental study by Weber et al. [50] is taken as the baseline configuration to be optimized.

3.1 Surrogate Model Optimization for Rosenbrock Function

The surrogate model optimization is validated by applying optimization process to the Rosenbrock test function which is used to test optimization algorithms and has a definite optimum point. The first step in optimization process is the defining the problem which is "optimizing $X_1 \in [-4,4]$ and $X_2 \in [-4,4]$ to find minimum point of the Rosenbrock function". $X1$ and $X2$ are input variables of the Rosenbrock functions, and it can be formulized as follows:

$$f(X1, X2) = (1 - X1)^2 + 100(X2 - X1^2)^2$$

The initial number of samples is determined in accordance with the rule of thumbs [24] which is defined as the number of variables multiplied by 10, which corresponds to 20. Using LHS method to initialize 20 samples for Rosenbrock function is shown in Figure 3.2.

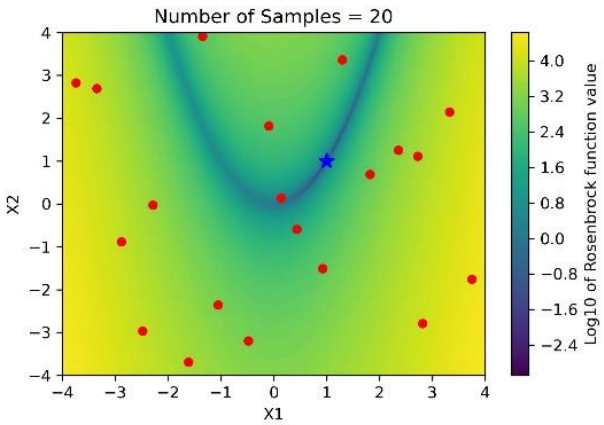


Figure 3.2 Initial samples of Rosenbrock functions and optimum point

The Kriging surrogate model is a mathematical model that can constitute the relation between input and output. It needs variables and results of samples to build a surrogate model. Figure 3.2 demonstrates the distribution of samples over the design space, Figure 3.3 and Figure 3.4 show that the real values of Rosenbrock functions for initial samples and dashed lines corresponds to optimum point after first iteration which only uses initial samples to find the optimum point. As can be seen from figures, despite the minimum value of the Rosenbrock function for initial samples is equal to nearly 20, optimization tool finds the optimum point value as 0.9914.

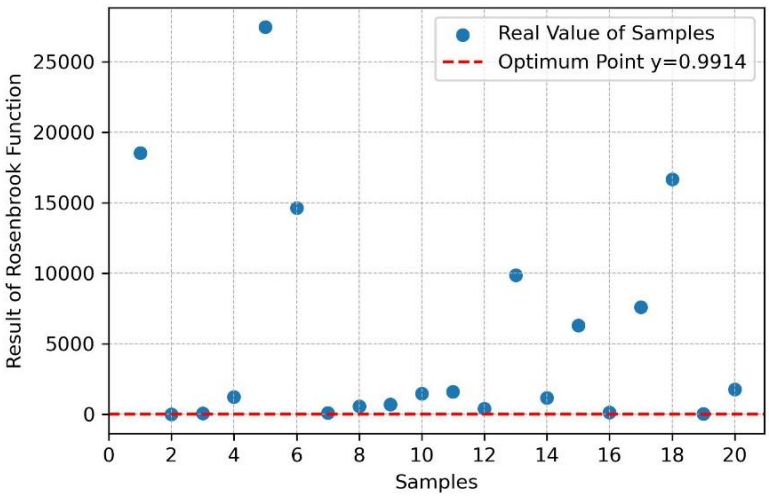


Figure 3.3 Initial samples' results of the Rosenbrock functions in full range

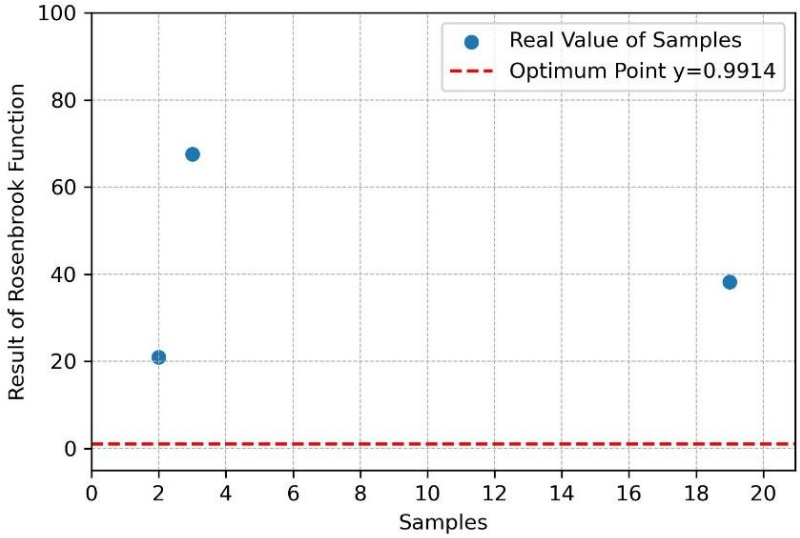


Figure 3.4 Initial samples' results of the Rosenbrock functions f(x) ∈ [0,100]

The Kriging surrogate model is validated by the LOOCV methodology which is K-fold cross-validation method with test subset as one sample. R^2 and RMSE are calculated to investigate the consistency of the model as performance parameters. R^2 score (R^2) for n samples is formulized as follows (y_i is the real value of Rosenbrock function, $\hat{y}_i$ is the predicted value coming from the Kriging surrogate model, $\overline{y}_i$ is mean of the real value of the Rosenbrock function):

$$R^2 = 1 - \frac{SS_{residual}}{SS_{total}} = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)}{\sum_{i=1}^{n}(y_i - \overline{y}_i)}$$

RMSE for n samples are formulized as follows (y_i is the real value of Rosenbrock function, $\hat{y}_i$ is the predicted value coming from Kriging surrogate model):

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

The first criteria to select the suitable model can be defined as the change in RMSE stays nearly constant with the number of samples. Change of RMSE with the number of samples is shown as Figure 3.5. The consistency of the model ensures that the

model is reliable to use in optimization process. As mentioned before, the optimum point coming from the design cycle is taking into the samples for the next step. Decreasing RMSE from Figure 3.5 also shows that incremental sampling method works properly.

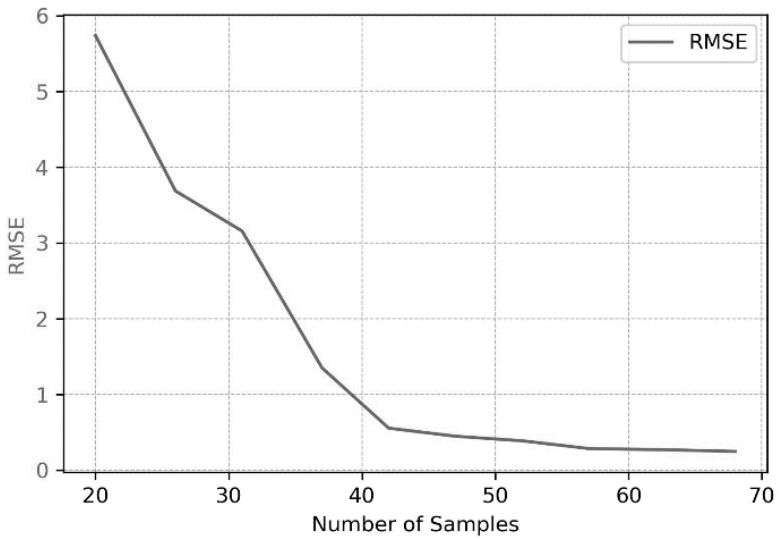


Figure 3.5 Variation of RMSE with number of samples

Figure 3.6 shows how the samples are distributed over the iterations. As expected from the algorithms, the homogeneity in the space is increasing over the iterations. Instead of increasing the consistency of the model in the local region, it improves all over the space, so that it minimizes the chance of missing a global optimum point.

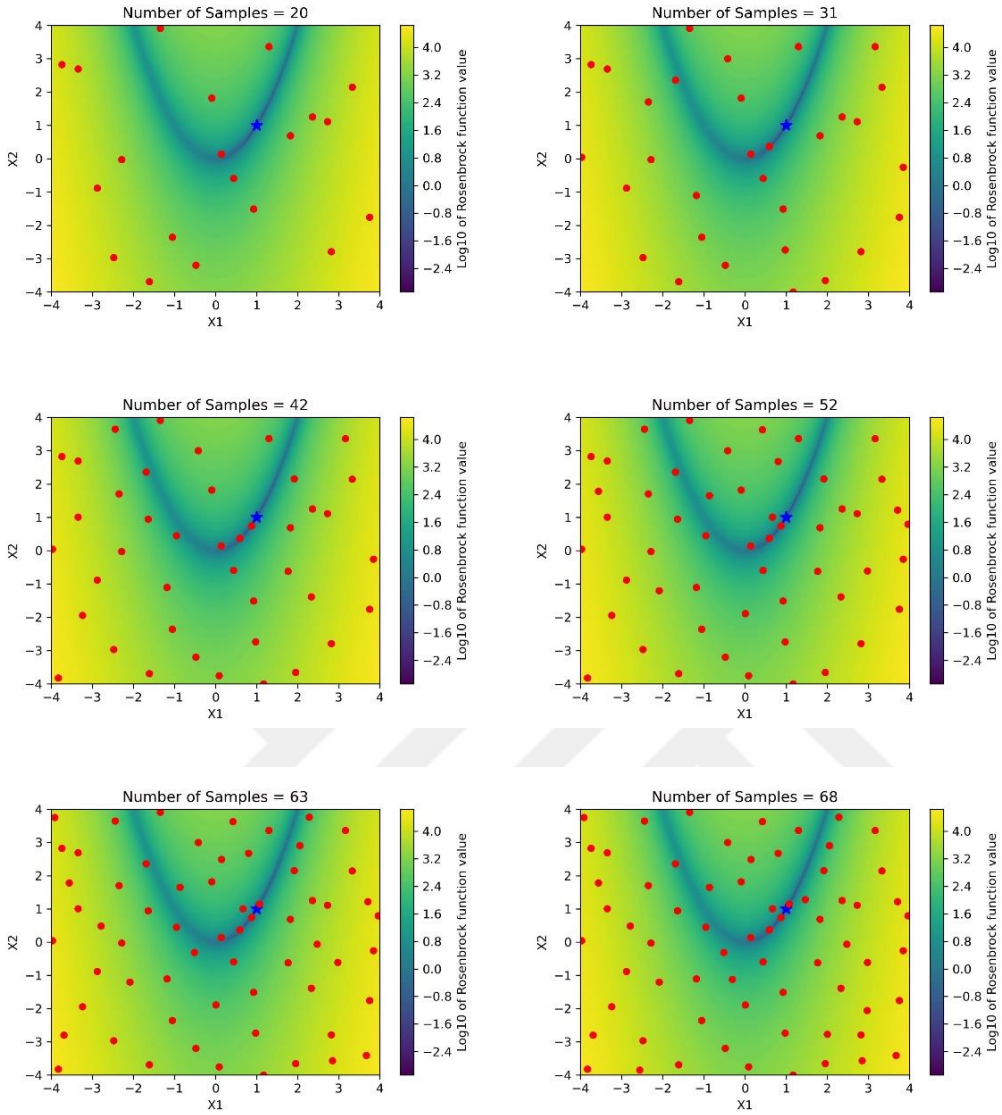


Figure 3.6 Distribution of the sample over the iterations

The R^2 score is another well-known parameter which is used in statistics to measure the consistency of the model regarding the real function. R^2 being close to 1 means that the results coming from the surrogate model and the real function are close to each other.

Table 3.1 demonstrates that R^2 is nearly 1 in all models. Figure 3.7 demonstrates the closeness of the value of the real functions and the results coming from LOOCV. It is also consistent with R^2 values. R^2, which is already close to 1, is sometimes not

enough to explain the accuracy of the real and the model results. The most common ones for this situation are large design space or high variation in the real functions. It is advised to check other error parameters such as RMSE. In addition to R^2, it is better to investigate RMSE as a more sensitive and accurate parameter to investigate the consistency of the model which is given in Figure 3.5. Deviation of points from black solid line, as shown in Figure 3.7, is expected scatter less for the accurate model. The number of samples coming from the incremental sampling method is defined as 5 and one sample is coming as a result of the optimization process. For the repeated optimum point, the interval is 5; otherwise, it is equal to 6, which can be seen from Table 3.1.

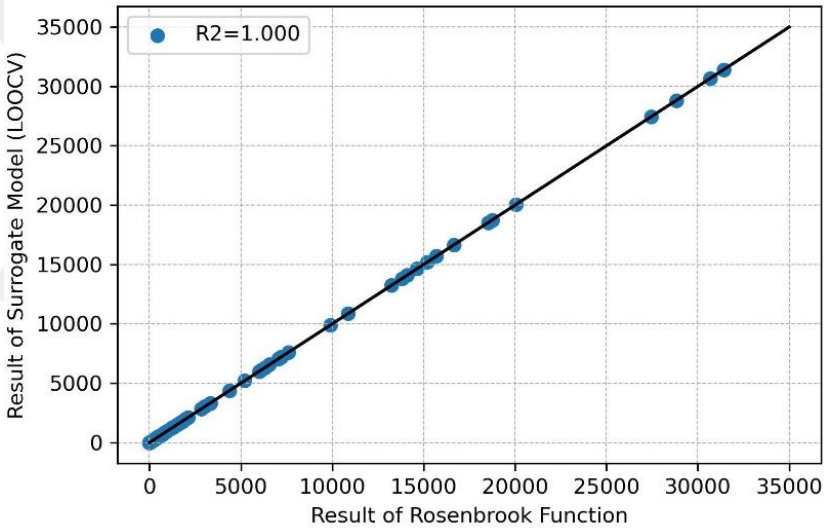


Figure 3.7 Comparison results of surrogate model (LOOCV) and Rosenbrock functions

Table 3.1 Results of surrogate model and numerical simulation and R^2, RMSE (F_m is result of surrogate model, F_r is result of Rosenbrock function)

# of samples	$X_{opt} = [X1, X2]$	$F_m(X_{opt})$	$F_r(X_{opt})$	R^2	$RMSE$
20	[1.9956, 3.9840]	-0.5087	0.9914	0.999993	5.781
26	[0.7403, 0.5527]	0.1172	0.0695	0.999999	3.692
31	[0.7403, 0.5527]	0.1632	0.0695	1.000000	3.142
36	[0.7403, 0.5527]	0.1772	0.0695	1.000000	1.417
42	[0.9449, 0.9059]	0.0070	0.0020	1.000000	0.628
53	[0.9853, 0.9670]	-0.0400	0.0016	1.000000	0.439
58	[0.9853, 0.9670]	-0.0387	0.0016	1.000000	0.357
63	[0.9853, 0.9670]	-0.0345	0.0016	1.000000	0.302
68	[0.9853, 0.9670]	-0.0028	0.0016	1.000000	0.287

3.2 Surrogate Model Optimization for Delta Wing

The primary usage of the surrogate model optimization is to bring the high-fidelity numerical tools in the design process and to decrease the required design time. FLOWUnsteady, a meshless open-source CFD tool, is selected as a high-fidelity numerical tool in the design of delta wing. Baseline delta wing configuration, which is taken from the experimental studies of Weber et al. [50] is also validated by the study of Alvarez [48]. Flow conditions and geometric parameters of the baseline configuration are shown in Figure 3.8.

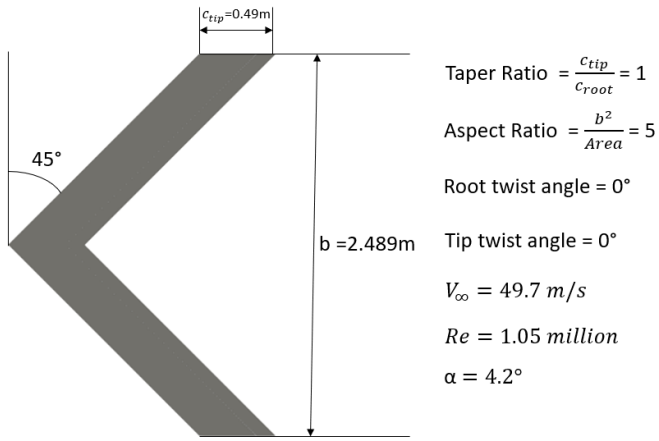


Figure 3.8 Flow conditions and geometric parameters of baseline configuration

The baseline configuration is also one of the validation test cases represented in the study of Alvarez [48]. Results coming from FLOWUnsteady and the experiment of Weber et al. [50] are given in Table 3.2. The optimization problem is defined as optimizing geometric variables to maximize (C_L/C_D) in this study.

Table 3.2 Comparison of numerical simulation and experimental results

Test Condition	C_L	C_D	C_L/C_D
Numerical Solution Result	0.238	0.005	48.57
Experimental Result	0.2351	0.00498	47.02
Percentage Error	1.2%	0.4%	3.2%

Various variables are chosen as input to investigate the functionality of the optimization tool. In the first case, root and tip twist angles are selected as variable parameters to maximize (C_L/C_D). In the second one, taper ratio, aspect ratio and sweep angle are chosen as variables in the design cycle.

3.2.1 Optimization of Delta Wing with Twist Angles

In the first case, root and tip twist angles are selected as variable parameters to maximize (C_L/C_D). Twist angles directly affect the vortex shading behind the wings which is not easy to capture with empirical or semi-empirical methods. It is recommended to use CFD tools to capture this phenomenon. Demonstration of root and tip twist angles (θ_{root} and θ_{tip}) are shown in Figure 3.10. Root and tip twist angles are swept to find the maximum (C_L/C_D) in the design space which is given in Table 3.3.

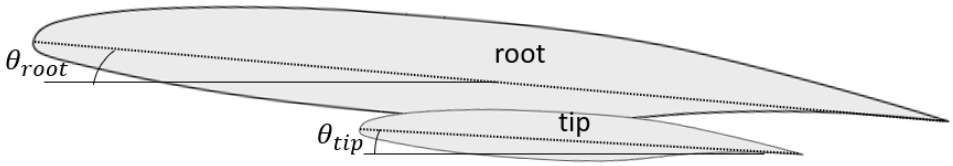


Figure 3.9 Demonstration of wing twist angles

Table 3.3 Design space of twist angles

Variable	*Design Space*
Root Twist Angle (θ_{root})	-5° to 5°
Tip Twist Angle (θ_{tip})	-5° to 5°

The initial number of samples, shown in Figure 3.10, is determined in accordance with the rule of thumbs [24] which is the number of the variables multiplied by 10, corresponding to 20. Initial samples are specified by LHS.

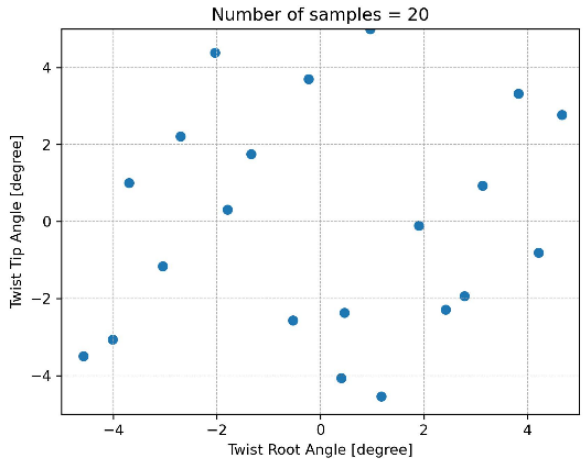


Figure 3.10 Initial sampling with LHS

Optimization process with initial sampling points is represented in Figure 3.11 which includes CFD results of samples, the optimum point after 1st iteration and the baseline configuration. CFD results of samples form a scattered distribution of various values in the design spaces. Figure 3.11 shows that the first iteration has already improved the objective function (C_L/C_D) according to the baseline configuration.

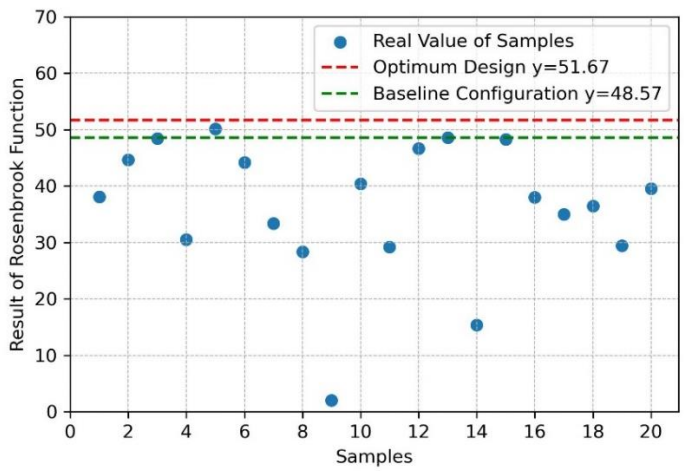


Figure 3.11 CFD results (C_L/C_D) of initial samples and optimum design

The number of the initial sampling points and the increment coming from the incremental sampling method in each iteration are defined as 20 and 5, respectively.

In addition to that, optimum design is also added to the following iteration's design space. As mentioned, the incremental sampling method is not only focusing on the optimum region, but also increasing the density of sampling over the design space. It helps to catch up the global optimum points and avoid being stuck with the local optimum point. Initial sampling and incremental sampling along different iterations are shown in Figure 3.12

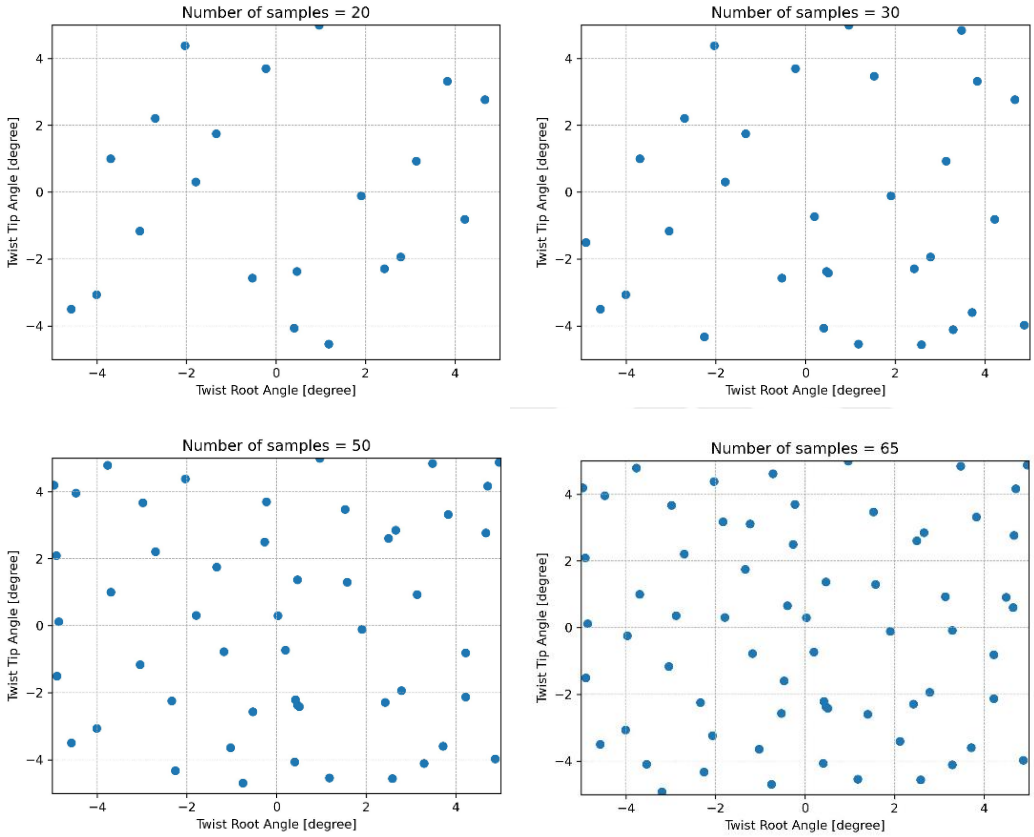


Figure 3.12 Initial sampling and increment samples

R^2 and RMSE values are calculated to investigate the consistency of the model as performance parameters. R^2 for n samples are formulized as follows (y_i is results of CFD, $\hat{y}_i$ is predicted value coming from Kriging surrogate model, $\overline{y_i}$ is mean of results of CFD):

$$R^2 = 1 - \frac{SS_{residual}}{SS_{total}} = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)}{\sum_{i=1}^{n}(y_i - \overline{y_i})}$$

RMSE for n samples are formulized as follows (y_i is results of CFD, $\widehat{y_i}$ is predicted value coming from Kriging surrogate model):

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{n}(y_i - \widehat{y_i})^2}$$

The maximum number of iterations is limited to 10 and no stopping criteria is defined to follow up the behavior of the convergence and the change of the optimum points. Regarding all samples and results, the mean of data is 35.6 and standard deviation is 12.47. Variation of RMSE with respect to the number of samples is shown in Figure 3.13. RMSE value can be evaluated as lower, which is less than ≈1.5% of 3σ. It means that the consistency of the surrogate model is reliable enough to trust the results.

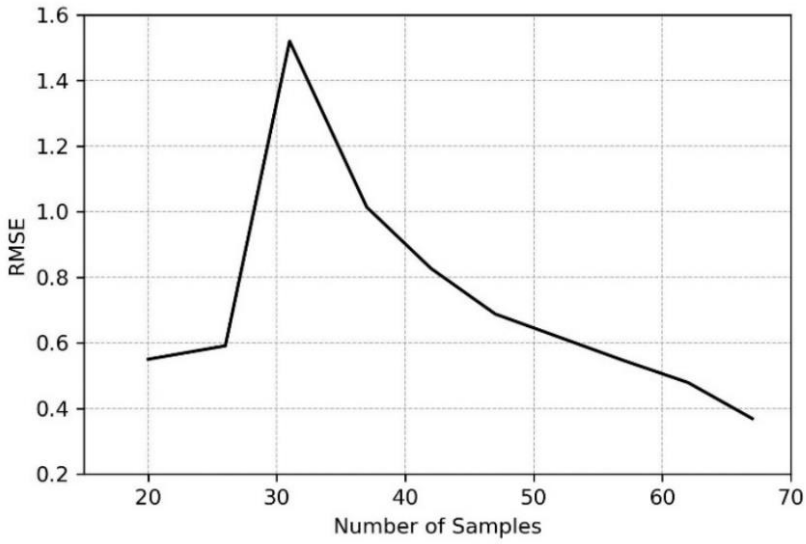


Figure 3.13 RMSE investigated from LOOCV of surrogate model

R^2, which also represents the consistency of both the surrogate model and CFD, are tabulated in Table 3.4. Comparison of the cross-validation results and CFD analysis is demonstrated in Figure 3.14. Closeness of R^2 to 1 and small differences between

the surrogate model and CFD results which are tabulated in Table 3.4 support the trustworthiness of the results of the surrogate model.

Table 3.4 Results of surrogate model and numerical simulation and R^2, RMSE (F_m is result of surrogate model, F_{CFD} is result of CFD, θ is twist angle)

# of samples	$X_{opt} = [\theta_{root}, \theta_{tip}]$	$F_m(X_{opt})$	$F_{CFD}(X_{opt})$	R^2	$RMSE$
20	[0.4628°, -2.3741°]	51.67	51.78	0.997	0.550
26	[0.4628°, -2.3741°]	51.78	51.78	0.998	0.591
31	[0.5029°, -2.4188°]	51.78	51.75	0.997	1.520
37	[0.4186°, -2.2099°]	51.80	51.81	0.975	1.014
42	[0.4186°, -2.2099°]	51.81	51.81	0.987	0.827
47	[0.4186°, -2.2099°]	51.81	51.81	0.992	0.688
52	[0.4186°, -2.2099°]	51.81	51.81	0.995	0.617
57	[0.4186°, -2.2099°]	51.81	51.81	0.996	0.546
62	[0.4186°, -2.2099°]	51.81	51.81	0.997	0.479
67	[0.4186°, -2.2099°]	51.81	51.81	0.998	0.369

The behavior of the numerical simulation can be varied in accordance with the variables and the design space. In this case, many local optimum points are very close to the global optimum point, which can also be realized from Table 3.4. The difference between CFD tool and the surrogate model can be more than the difference between local and global optimum points. For such a situation, although the surrogate model comes with different optimum points, results of the real function in local and global optimum points are expected to get closer to each other. It is concluded that the optimization process leads to efficient results. Small decrement in the 3rd iteration is an example of this phenomenon. As R^2 highly depends on the location of the sampling points, adding new points to the design space causes fluctuation in R^2 values. Decrement in R^2 is caused by the change of the confidence level resulting in adding new points.

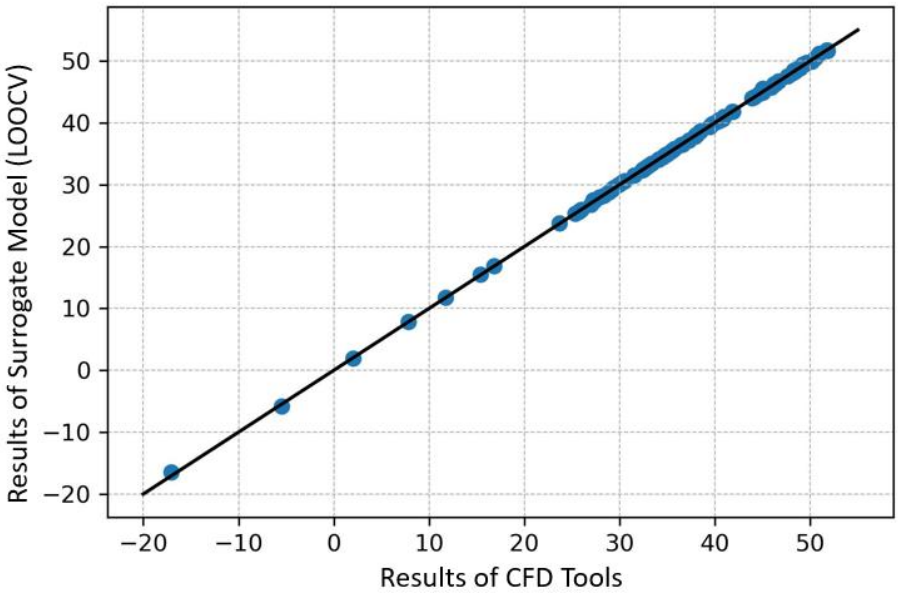


Figure 3.14 Comparison results of CFD and surrogate model (LOOCV)

The configuration which is performed in the wind tunnel test has zero tip and root twist angle. The CFD results of the wind tunnel test configuration are given in Table 3.5. The objective function, which is (C_L/C_D), is being tried to be maximized in the optimization problem. The optimum point found by the surrogate model built in the 1st iteration with initial sample points is [0.4628, -2.3741]. The (C_L/C_D) coming from the surrogate model for the optimum point is 51.68 and the result coming from CFD tool is 51.77. The small difference between these two results also proves that the surrogate model accurately represents the CFD tool. After the optimization loop (C_L/C_D) value increased by 6.6%. After the first iteration, the change of (C_L/C_D) in the successive optimum points does not vary significantly. Indeed, in some iterations, optimum points remain the same. This means that no new optimum point is found in these iterations, or global optimum point is found in this iteration. CFD results of the baseline and the optimum design configurations and their comparison are tabulated in Table 3.5.

Table 3.5 Comparison result of optimum and baseline design

Configuration	C_L	C_D	C_L/C_D
Baseline (CFD Solution) $\theta_{root} = 0°, \theta_{tip} = 0°$	0.238	0.005	48.57
Optimum Design (CFD Solution) $[\theta_{root} = 0.4186°, \theta_{tip} = 2.2099°$	0.1974	0.003811	51.77
Percentage Difference	-17%	-23.8%	6.6%

All these configurations are also solved by unsteady flow solver which is FLOWUnsteady. Optimum design configuration of the C_L and C_D history over the time is represented in Figure 3.15 and the converged solution is reached in 0.14 seconds of the simulation. Results of the optimum design, baseline and their comparison are tabulated in Table 3.5. In addition to that, the solver has sectional solutions in the spanwise which are shown in Figure 3.16. Red lines and blue lines represent the solution over time and the final solution corresponding to 0.14 seconds, respectively.

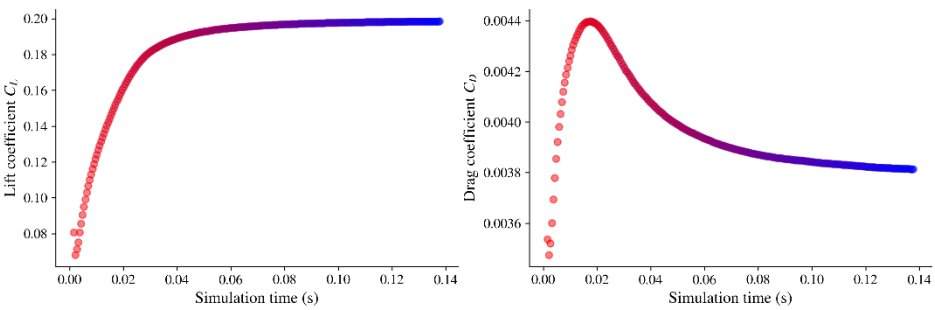


Figure 3.15 C_L and C_D at the optimum configuration

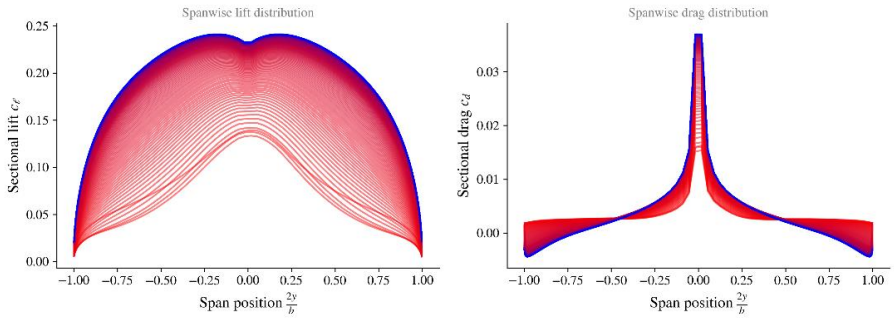
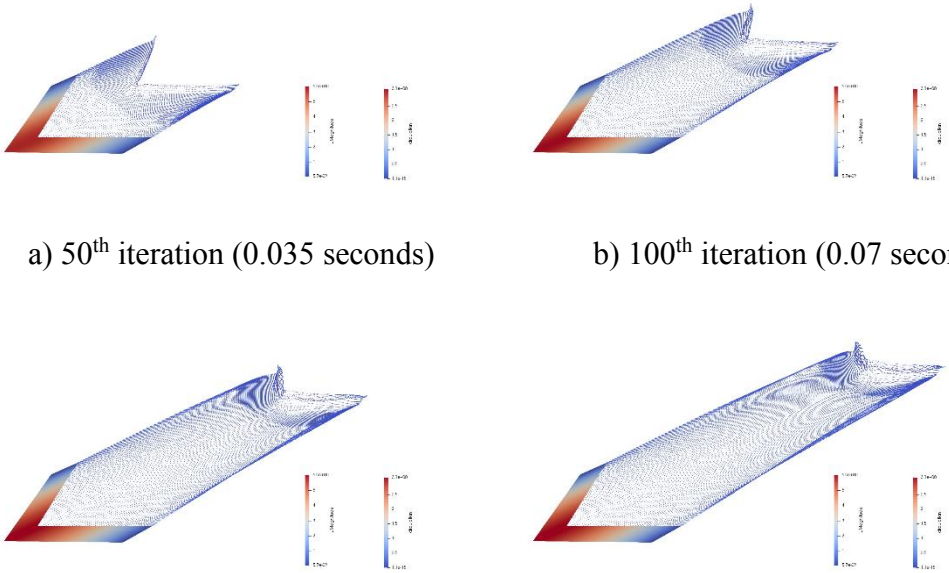


Figure 3.16 Spanwise lift and drag coefficient

Instantaneous wake developments are shown with particle distributions in different times. The motion of particles over the simulation time are demonstrated in Figure 3.17. The motion of the particles for different time periods are 50th iteration (0.035 seconds), 100th iteration (0.07 seconds), 150th iteration (0.105 seconds) and 200th iteration (0.14 seconds). The simulation is converged when the iteration reaches 200 which corresponds to 0.14 seconds.



a) 50th iteration (0.035 seconds)

b) 100th iteration (0.07 seconds)

c) 150th iteration (0.105 seconds)

d) 200th iteration (0.14 seconds)

Figure 3.17 Instantaneous wake developments shown with particle distributions in different times

3.2.2 Optimization of Delta Wing with Aspect Ratio, Taper Ratio and Sweep Angle

The first optimization process, in which the twist angles are defined as variables, results in 6.5% enhancement. In this design case, aspect ratio, taper ratio and sweep angle are used to find the optimum point of the design space which is represented in Table 3.6. The schematic view of aspect ratio, taper ratio and sweep angle are shown in Figure 3.18. As the behavior of the variables on the objective function, (C_L/C_D), can be changed, using different variables in the design tool also gives insight about the capability of the tool. In addition to that, the optimization tool is also tested with three variables. As in the previous case, the aim is to maximize the (C_L/C_D) in the design space.

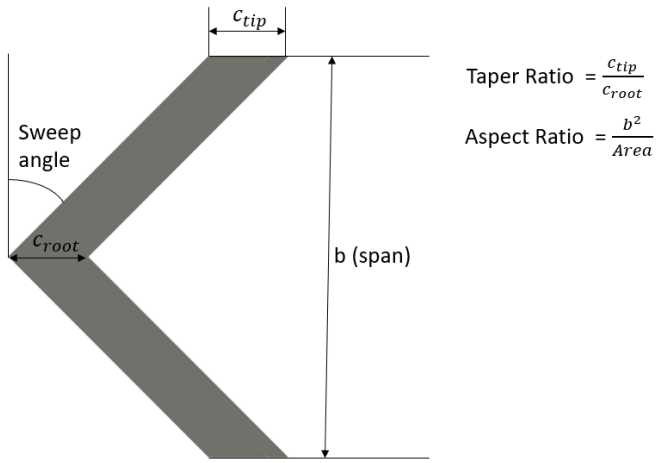


Figure 3.18 Demonstration of design parameters

Table 3.6 Design space of optimization of delta wing

Design Variable	*Design Space*
Aspect Ratio	[2, 6]
Taper Ratio	[0.1, 1]
Sweep Angle	[0°, 60°]

The initial number of samples, shown in Figure 3.19, is determined according to rule of thumbs [24] which is the number of variables multiplied by 10, corresponding to 30. Initial samples are specified by LHS.

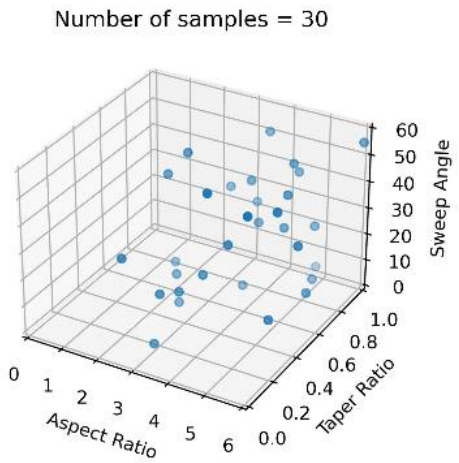


Figure 3.19 Initial sampling with LHS

Aspect ratio, taper ratio and sweep angle are the fundamental geometric parameters to create a wing shape. Some of the random configurations, selected from the design space, are demonstrated in Figure 3.20. The aspect ratio, taper ratio and sweep angle can construct different varieties of the configuration from the conventional wing design to the high sweep angle delta wings.

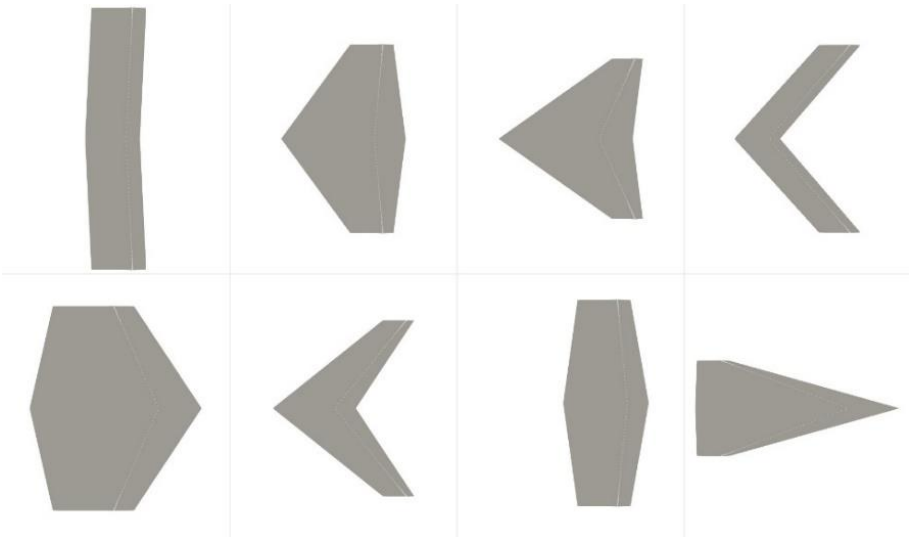


Figure 3.20 Different configurations from design space of delta wing

The optimization process with the initial sampling points is represented in Figure 3.21 which contains the CFD results of the samples, the optimum point after 1st iteration and the baseline configuration. The CFD results of the samples are distributed as scattering various values in the design space. Figure 3.21 shows that 1st iteration has already improved the objective function (C_L/C_D) according to the baseline configuration.

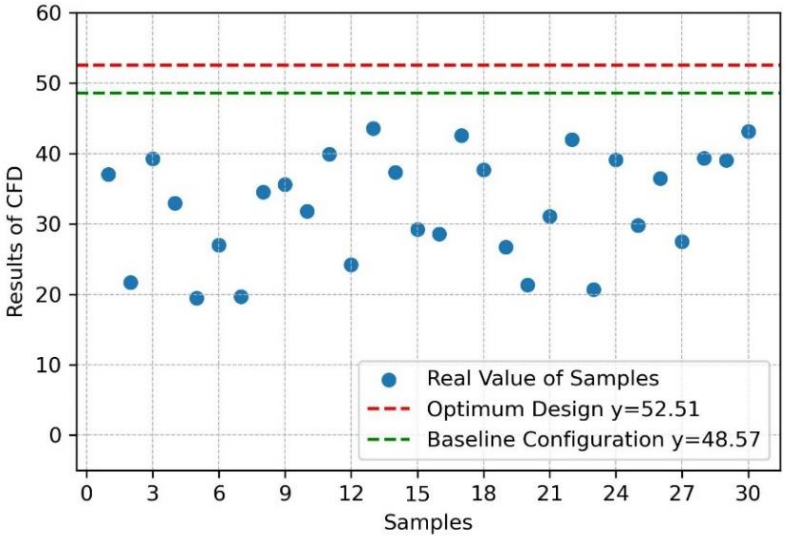


Figure 3.21 CFD results (C_L/C_D) of initial samples and optimum design

The number of initial sampling points and increment in each iteration are defined as 30 and 5, respectively. The initial sampling distribution over the design space and the change of distribution over iterations are demonstrated in Figure 3.22. LHS methodology is a very effective method to combine randomness and even distribution for each variable. On the other hand, it is not easy to forecast the required number of samples in the beginning of the process. Incremental sampling method, which is based on increasing the number of samples in the optimum region, sometimes results in missing the global optimum points. The incremental sampling method used in this study not only increases the samples around the optimum points, but also keeps the homogeneity in the design space. It can easily be seen from Figure 3.22 that the homogeneity is satisfied with increasing number of samples. This idea

is also supported by the change of the R^2 with the number of samples which are tabulated in Table 3.7.

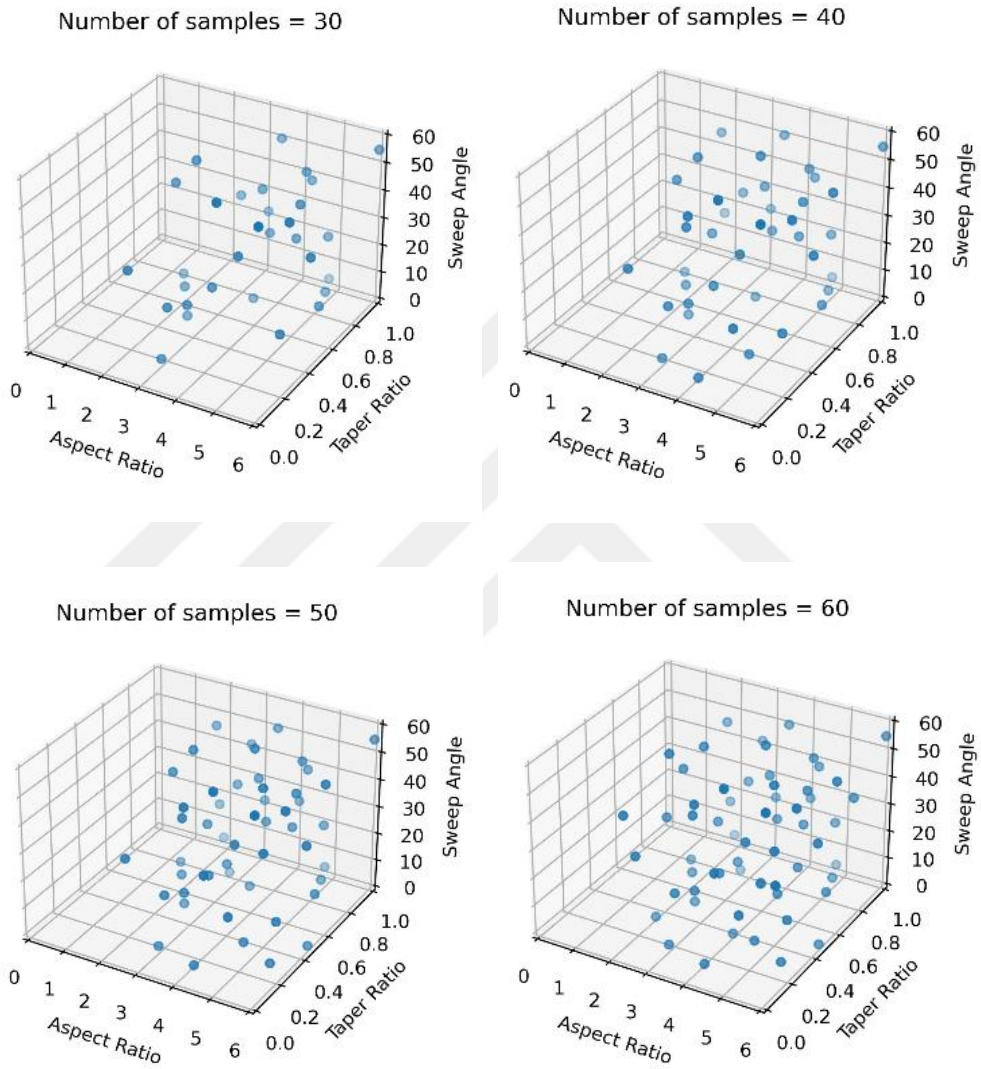


Figure 3.22 Distribution of the samples over the iterations

R^2 and RMSE are calculated to investigate the consistency of the model as performance parameters.

The run is terminated in the 8th iteration since the optimum point, which is found in the first iteration, does not change. On the other hand, the iterative process is continued to be sure about the consistency of the surrogate model and CFD results. Regarding all samples and results, the mean of the data is 32.5 and the standard deviation is 7.83. Variation of RMSE with respect to the number of samples are shown in Figure 3.23. RMSE value can be evaluated as lower level, which is around 1% of 3σ. It means that the consistency of the surrogate model is reliable enough to trust the results.

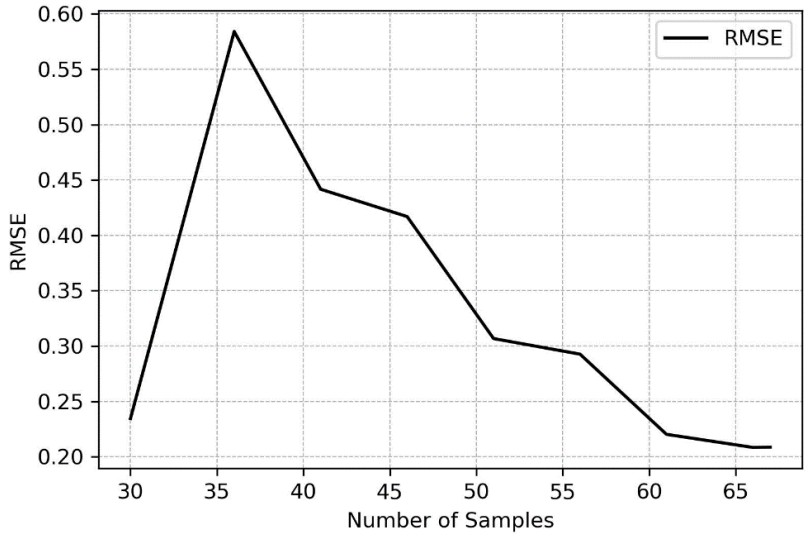


Figure 3.23 RMSE of surrogate model with LOOCV

The surrogate model is just a mathematical model which has no physical background, and it has no capability of evaluating whether the data is logical. In the dynamic optimization cycles, defining input, running numerical solution, adding points to surrogate model and finding optimum points are generated and worked without human control. It is very critical to decide whether the results coming from numerical solutions are logical. Otherwise, it is possible that the process concludes with wrong results. It is needed to control the data coming from numerical solutions and it can be achieved by checking whether the results are in the logical interval, which is determined as [-5 5] for both variable C_L and C_D in this case.

The R^2, which is tabulated in Table 3.7, is also a widely used parameter to measure consistency of the model. The R2 value can be evaluated as the deviation of the cross validation and results of the real functions from 45° inclined line which is shown in Figure 3.24. The optimum point found right after the 1st iteration does not change the rest of the process which shows that the initial number of the samples is enough to find the optimum point. The optimum point and the result of model do not change along the iterations, in contrast to that, R^2 becomes closer to 1, which means that the surrogate model becomes a better approximation of CFD tool while number of iteration increases.

Table 3.7 Results of surrogate model and numerical simulation and R^2, RMSE (F_m: Result of surrogate model, F_{CFD}: Result of CFD, AR: Aspect ratio, TR: Taper ratio, Λ: Sweep angle)

# of samples	$X_{opt} = [AR, TR, \Lambda]$	$F_m(X_{opt})$	$F_{CFD}(X_{opt})$	R^2	$RMSE$
30	[6.0000, 0.9967, 54.8918°]	50.83	52.51	0.9989	0.242
36	[6.0000, 0.9967, 54.8918°]	52.51	52.51	0.9942	0.581
41	[6.0000, 0.9967, 54.8918°]	52.51	52.51	0.9969	0.439
46	[6.0000, 0.9967, 54.8918°]	52.51	52.51	0.9971	0.423
51	[6.0000, 0.9967, 54.8918°]	52.51	52.51	0.9983	0.309
56	[6.0000, 0.9967, 54.8918°]	52.51	52.51	0.9992	0.289
61	[6.0000, 0.9967, 54.8918°]	52.51	52.51	0.9992	0.221
66	[6.0000, 0.9967, 54.8918°]	52.51	52.51	0.9993	0.208

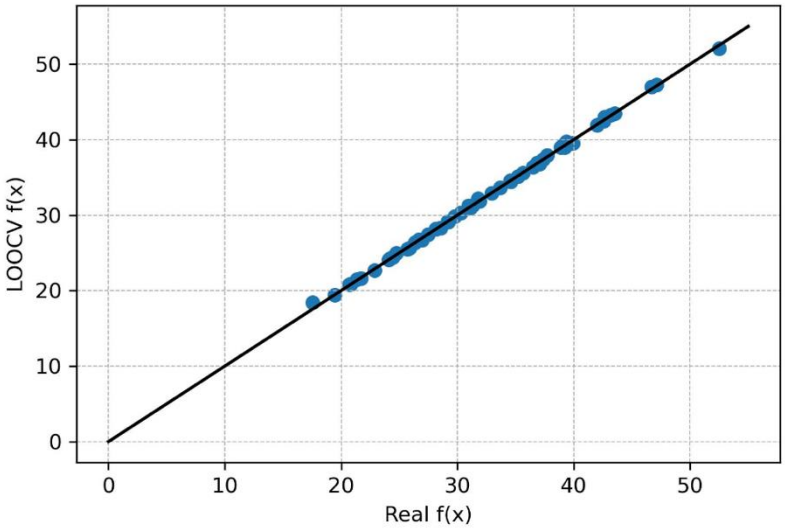


Figure 3.24 Comparison of cross-validation and real functions results

The configuration which is tested in the wind tunnel is taken as a baseline and geometric parameters are shown in Figure 3.25. The results for wind tunnel test configuration and the optimum design for C_L, C_D and (C_L/C_D) are compared in Table 3.8. The objective function, which is (C_L/C_D), is tried to be maximized in the optimization problem. In the design space, (C_L/C_D) value increased by 8.1% with the optimum design and its geometric parameters which are also demonstrated in Figure 3.25. After the 1st iteration, the value of (C_L/C_D) in the successive optimum points does not change. Generally, this phenomenon is seen for the variables which have dominant linear trend on the real function which is the relation between (C_L/C_D) and the aspect ratio, taper ratio and sweep angle.

Table 3.8 Comparison results of optimum and baseline design (AR: Aspect Ratio, TR: Taper Ratio, Λ: Sweep Angle)

Configuration	C_L	C_D	C_L/C_D
Baseline (CFD Solution) $AR = 5, TR = 1\ \Lambda = 45°$	0.238	0.005	48.57
Optimum Design (CFD Solution) $[AR = 6, TR = 0.996\ \Lambda = 54.9°$	0.2138	0.00407	52.51
Percentage Difference	-10%	-18.6%	8.1%

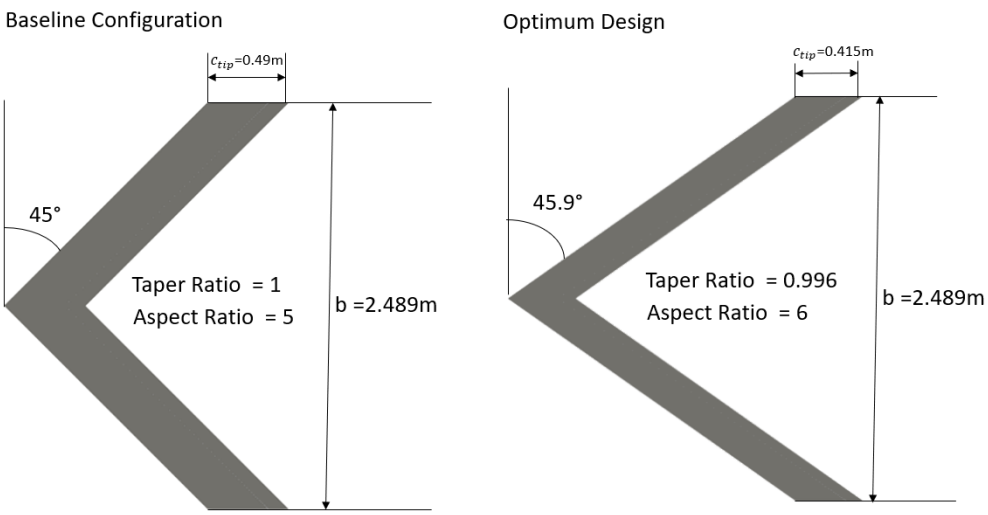


Figure 3.25 Baseline (left) and optimum design (right) configurations

The optimum configuration is also solved by unsteady solver which is FLOWUnsteady like the other configurations. The $C_L$and C_D history over the time is represented in Figure 3.26. All solutions are converged by 200 iterations which correspond to 0.14 seconds. At the optimum point, C_L, C_D and (C_L/C_D) are calculated as 0.2138, 0.004071 and 52.5115 respectively. In addition to that, the solver has sectional solutions in the spanwise which are shown in Figure 3.27. Red lines

represent solution over iterations and blue line represents the final solution which corresponds to 0.14 seconds.

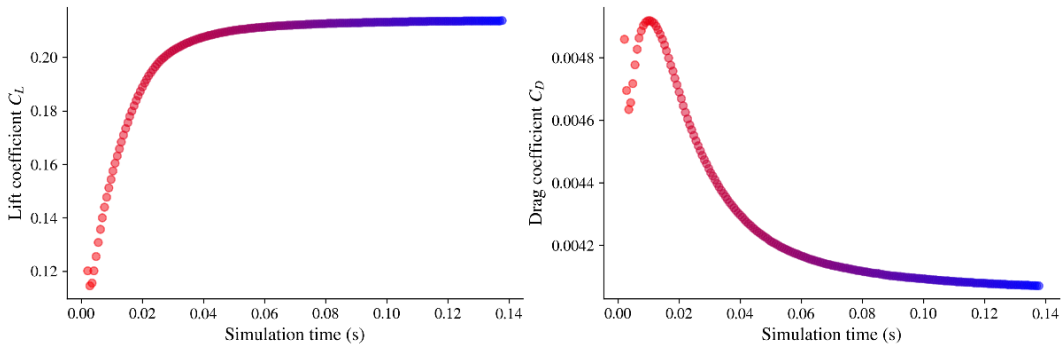


Figure 3.26 C_L and C_D of the optimum design

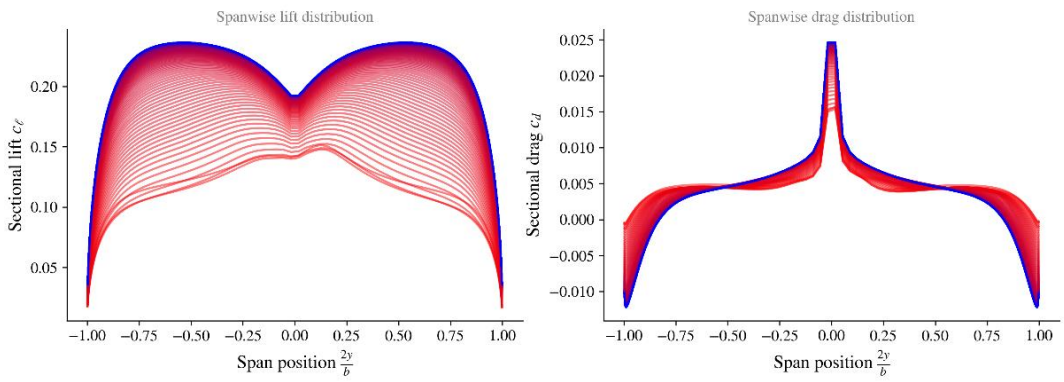
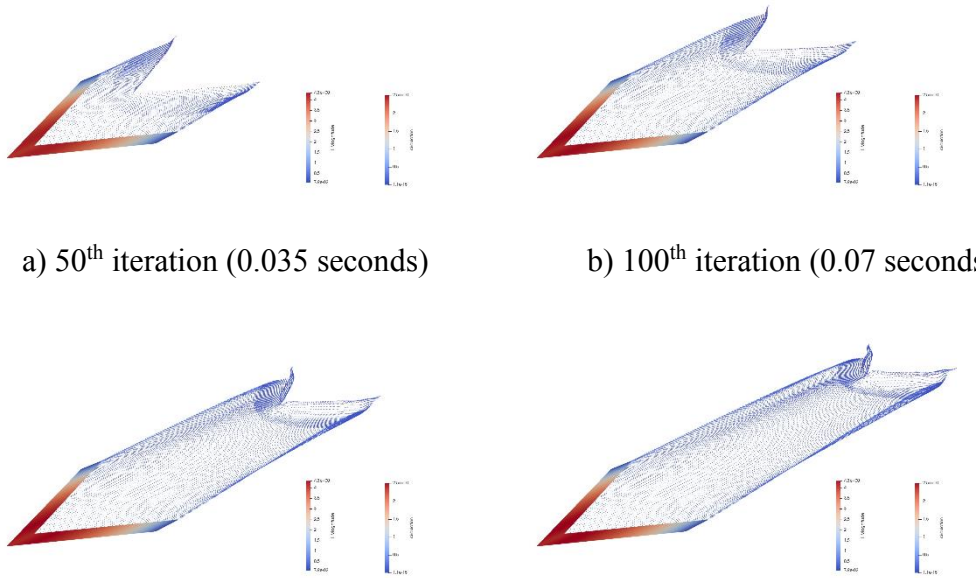


Figure 3.27 Spanwise lift and drag coefficients of the optimum design

The best demonstration of the unsteady solver for VPM, which is a mesh-free CFD approach solving the Navier-Stokes equations in their velocity-vorticity form, is the motion of particles over the simulation time. The time when particles' motion does not change corresponds to the convergence time of the simulation. Motions of the particles for different time periods, which are 50th iteration (0.035 seconds), 100th iteration (0.07 seconds), 150th iteration (0.105 seconds) and 200th iteration (0.14 seconds) are demonstrated in Figure 3.28. With an enough extension of the wake downstream at around 0.14 seconds as shown in Figure 3.28, the solutions for the lift and drag coefficients as shown in Figure 3.26 and similarly the spanwise load distributions as shown in Figure 3.27, converge to the steady-state values.



a) 50th iteration (0.035 seconds) b) 100th iteration (0.07 seconds)

c) 150th iteration (0.105 seconds) d) 200th iteration (0.14 seconds)

Figure 3.28 Instantaneous wake developments shown with particle distributions in different times

In this study, an optimization tool with hybrid incremental sampling methodology is developed. The noticeable feature of the optimization tool from the others is that incremental sampling methodology can keep features of LHS and also applicable to any number of samples. The results of the Rosenbrock test case and delta wing design optimizations prove the accomplishment of optimization tool to find optimum design. In addition to that, comparisons of the results, both initialization with LHS method and the results of the incremental sampling method are also investigated. In the delta wing design optimization, optimization cycle starts with 30 samples and reaches 51 samples in the 4th iteration which is called 30i51n run. Optimum points after 1st and 4th iterations are equal. Another optimization run is directly initialized with 51 samples which is called 51i51n run using LHS to compare the results with 30i51n run. Results of both runs, which are represented in Table 3.9, conclude with nearly same optimum points. Results of the objective function, (C_L/C_D), increases 0.3%, R^2 increases 0.04% and RMSE changes nearly 0.2% regarding mean of the

samples for 51i51n. Table 3.9 shows that 30i51n run reaches nearly the same results with 51i51n run results which is initialized with LHS. In contrast to small improvements in 51i51n run results, the optimization cycle conducted with 30i51n run reaches the optimum design with 30 samples which corresponds nearly 60% of total run time of 51i51n run. Moreover, it prevents starting with an insufficient number of initial samples. Insufficient number of initial samples results in inaccurate surrogate model and concludes with duplication of initial runs for LHS. The incremental sampling method gives chance to increase samples while keeping the initial samples with nearly the same performance parameters.

Table 3.9 Results of surrogate model and numerical simulation, and R^2 and RMSE (F_m: Result of surrogate model, F_{CFD}: Results of CFD, AR: Aspect ratio, TR: Taper ratio, Λ: Sweep angle, $n_{initial}$: number of initial samples)

# of samples	$X_{opt} = [AR, TR, \Lambda]$	$F_m(X_{opt})$	$F_{CFD}(X_{opt})$	R^2	$RMSE$
51 ($n_{initial}$ = 30) 30i51n run	[6.0000, 0.9967, 54.8918°]	52.51	52.51	0.9983	0.309
51 ($n_{initial}$ = 51) 51i51n run	[6.0000, 1.0000, 54.8918°]	52.93	52.67	0.9987	0.235
Percentage Difference	Optimum points are close	0.8%	0.3%	0.04%	-0.2%

The optimization cycle used in this study consists of DoE, numerical simulation construction of the surrogate model and optimization of the surrogate model. All components of the optimization cycle have different computational times which are given in Figure 3.29. The time durations are measured for an optimization iteration performing on Apple M1 chip to give an insight about the order of magnitude of the computational time. The computational time for the numerical simulation given in Figure 3.29 corresponds to a single run and depends on the fidelity of the

FLOWUnsteady [48]. As mentioned, inputs of all runs needed for one optimization iteration determined just after DoE, the numerical simulations for each part can be parallelized and duration the computational time duration for one optimization iteration can be shortened to nearly one numerical solution duration. As was presented in the first part of study, one optimization iteration leads to a better improvement. The part of the optimization cycle which has the longest time requirement is the numerical solution that represents the total time required to reach an optimum point. In classical optimization process, it is not possible to parallelize all runs, but it is only achievable to parallelize a single run. Parallelization of a single run has limitations coming from the hardware technology. In contrast to that, the optimization tool developed in this study has capability to parallelize not only a single run but also the total number of runs needed to reach an optimum point.

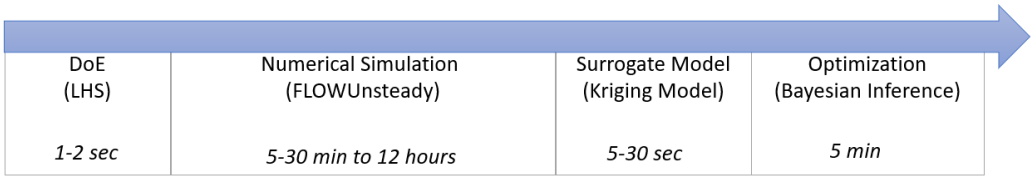


Figure 3.29 Computational time of optimization cycle

CHAPTER 4

CONCLUSION

Design optimization using a surrogate model consists of choosing samples in the design space, building surrogate model instead of using high-fidelity analysis tool and finding the optimum point in the design space using the stochastic approach.

In this study, a surrogate model-based optimization algorithm is developed and tested not only for the test function of Rosenbrock, but also for the selected delta wing design test case applications. In this thesis, surrogate model-based optimization and the motivation for an aerodynamic design process are introduced. Following that, historical background and related studies in the literature are presented. In the methodology section, DoE, Kriging surrogate model and Bayesian optimization methods are explained in detail.

Three different test cases, namely finding minimum point of Rosenbrock test function as a validation test case and two delta wing optimization test cases as an aerodynamic design application, are investigated to test the optimization algorithm and the parallel design strategy. In the first delta wing design case, the twist angle of the selected delta wing configuration is optimized. In the second delta wing design case, delta wing design based on different design parameters such as aspect ratio, taper ratio and sweep angle is performed. The results of test function and delta wing design cases show that the parallel design strategy has a good performance for reaching the optimum point.

The conclusions drawn from this thesis study can be summarized as below:

- The Kriging surrogate model is used as an effective stochastic method to model a high-fidelity numerical simulation, which are obtained by an unsteady flow solver FLOWUnsteady and the Bayesian inference

successfully finds the optimum point for the selected test function and the delta wing design optimization cases.

- Increasing the number of samples in the design algorithm, which is used to determine samples in each iteration, is suitable for parallelization. The parallel design strategy demonstrates that, with iterations, design time decreases and the consistency of the surrogate model with the real mathematical functions (i.e., exact solutions) increases. Although this strategy may increase the total CPU time to find an optimum point, the parallelization of the runs results in a decrease in design time.
- The results of test function show that the design algorithm works properly. This can be observed from the R^2 score and the difference between the numerical simulation and the surrogate model results.
- For the twist angle optimization case of delta wing design, the optimum point is found in the 3rd iteration. For the multi-parameter design case, i.e., aspect ratio, taper ratio and sweep angle optimization case for delta wing, the design algorithm finds the optimum point in the 1st iteration.
- The optimization algorithm works well for the multi-parameters design cases with two and three inputs.

The future work regarding this thesis may be recommended as follows:

- This optimization study can be enlarged by including more related design parameters such as aspect ratio, taper ratio, sweep angle, twist angle, and incidence angle and can be conducted for different flow regimes. In addition, interdisciplinary design test cases can be investigated.
- The hybrid parallel optimization tool developed in this study can also be used together with higher fidelity numerical simulation tools such as Navier-Stokes equations based CFD flow solvers or Lattice Boltzmann flow solvers.
- Different parallel design strategies can be implemented on the design code developed and hybrid algorithms can be studied. The Bayesian inference algorithm decides the following design point by using acquisition functions.

Although the most popular acquisition function is the expected improvement (EI); there are other functions such as knowledge-gradient, entropy search and upper confidence bound. The probability of improvement function are also used as acquisition functions in literature. Especially, the parallel design strategies may allow to choose more than one successive point which can be decided by combination of different acquisition algorithms' results, and such combinations are also called as hybrid algorithms.

REFERENCES

[1] G. Matheron, "Principles of geostatistics,," *Economic Geology, 58(8),,* p. 1246–1266, 1963.

[2] D. Krige, "A statistical approach to some mine valuation and allied problems on the Witwatersrand,," University of the Witwatersrand, Faculty of Engineering, Witwatersrand, 1951.

[3] R. Preece and J. V. Milanovic, "Efficient estimation of the probability of small-disturbance instability of large uncertain power systems," *IEEE Transactions on Power Systems, 31(2),* pp. 1063-1072, 2016.

[4] P. Frazier, W. Powell and S. Dayanik, " The knowledge-gradient policy for correlated normal beliefs," *INFORMS Journal on Computing, 21(4),* p. 599–613, 2009.

[5] J. Mockus, Bayesian approach to global optimization: Theory and applications,, New York: Kluwer Academic Publishers, 1989.

[6] D. R. Jones, M. Schonlau and W. J. Welch, "Efficient global optimization of expensive black-box functions," *Journal of Global Optimization, 13(4),* p. 455–492., 1998.

[7] A. I. Forrester, A. Sobester and A. J. Keane, "Multi-fidelity optimization via surrogate modelling," *Proceedings of the Royal Society of London A: Mathematical Physical and Engineering Sciences, 463,* p. 3251–3269, 2007.

[8] D. M. Negoescu, P. I. Frazier and W. B. Powell, "The knowledge gradient algorithm for sequencing experiments in drug discovery,," *INFORMS Journal on Computing, 23(1), ,* p. 46–363, 2011.

[9] P. I. Frazier and J. Wang, "Bayesian optimization for materials design," *Information Science for Materials Discovery and Design - Springer,* p. 45–75, 2016.

[10] D. Packwood, Bayesian optimization for materials science, Vol. 3, Springer., Berlin: Springer, 2017.

[11] D. Lizotte, "Practical Bayesian Optimization," PhD thesis, University of Alberta, Alberta, 2008.

[12] D. Lizotte, T. Wang, M. Bowling and D. Schuurmans, "Automatic gait optimization with Gaussian process regression," *Proceedings of IJCAI,* p. 944–949., 2007.

[13] J. Snoek, H. Larochelle and R. P. Adams, "Practical Bayesian optimization of machine learning algorithms,," *Advances in Neural Information Processing Systems,* p. 2951–2959, 2012.

[14] S. Toscano-Palmerin and P. I. Frazier, "Bayesian optimization with expensive integrands," *arXiv preprint,* p. arXiv:1803.08661., 2018.

[15] E. Dincer, "Aerodynamic design and performace analyses of grid fin in supersonic flow using design of experiments and computational fluid dynamics," METU, Ankara, 2022.

[16] E. Dincer and N. Sezer Uzol, "Parametric design and investigation of grid fin aerodynamics in supersonic flow using computational fluid dynamics,," in *AIAA AVIATION 2022 Forum*, Illinois, USA, 27 June - 01 July 2022.

[17] M. Gun, "External configuration design and aerodynamic optimization of modular guided munitions," METU, Ankara, 2019.

[18] M. Gun and N. Sezer-Uzol, "External Configuration Design and Optimization of Guided Munitions," in *AIAC-2019-108, 10th Ankara International Aerospace Conference, METU*, Ankara, TURKEY, 18-20 September 2019.

[19] K. Oguz, "Aerodynamic optimization of horizontal axis wind turbine blades by using CST method, BEM theory and genetic algorithm," METU, Ankara, 2019.

[20] K. Oguz and N. Sezer-Uzol, "Aerodynamic Optimization of Horizontal Axis Wind Turbine Rotor by Using BEM, CST Method and Genetic Algorithm," in *AIAC-2019-063, 10th Ankara International Aerospace Conference , METU*, Ankara, TURKEY, 18-20 September 2019.

[21] M. D. McKay, R. J. Beckman and W. J. Conover, "A 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.

[22] M. Liefvendahl and R. Stocki, "A study on algorithms for optimization of Latin hypercubes," *Journal of Statistical Planning and Inference, 136(9),* p. 3231–3247, 2006.

[23] B. Husslage, G. Rennen, E. R. Van Dam and D. den Hertog, "Space-filling Latin hypercube designs for computer experiments," *CentER Discussion Paper, No. 2006–18, Tilburg University, Tilburg, Netherlands.,* p. 18, 2006.

[24] J. S. Park, "Optimal Latin-hypercube designs for computer experiments,," *J. Stat. Plan. Inference, 39(1),* p. 95–111, 1994.

[25] A. Sóbester, S. Leary and A. Keane, "A parallel updating scheme for approximating and optimizing high fidelity computer simulations,," *Struct Multidisc Optim, 27,* p. 371–383, 2004.

[26] A. Mukhtar, A. S. H. Md Yasir and M. F. Mohamed Nasir, "A machine learning-based comparative analysis of surrogate models for design optimisation in computational fluid dynamics,," *Heliyon e18674,* p. 9(8), 2023.

[27] J. P. C. Kleijnen, "Kriging metamodeling in simulation: A review,," *Eur. J. Oper. Res., 192(3),* p. 707–716, 2009.

[28] J. Loeppky, J. Sacks and W. Welch, "Choosing the sample size of a computer experiment: A practical guide,," *Technometrics, 51,,* pp. 366–376, , 2009.

[29] J. P. C. Kleijnen, " Regression and Kriging metamodels with their experimental designs in simulation: A review," *Eur. J. Oper. Res., 256(1),* pp. 1-16, 2017.

[30] J. Sacks, W. J. Welch, T. J. Mitchell and H. P. Wynn, "Design and Analysis of Computer Experiments," *Statist. Sci. 4(4),* pp. 409-423, 1989.

[31] T. Simpson, T. Mauery, J. Korte and F. Mistree, "Kriging Models for Global Approximation in Simulation-Based Multidisciplinary Design Optimization," *AIAA J., 39,* p. 2233–2241, 2001.

[32] J. D. Martins and T. W. Simpson, "On Using Kriging Models as Probabilistic Models in Design," *SAE Trans., 113,* p. 129–139. , 2004.

[33] J. Martin and T. Simpson, "Use of Kriging Models to Approximate Deterministic Computer Models," *AIAA J., 43,* p. 853–863, 2005.

[34] G. Dellino, P. Lino, B. Zimanowski, R. Büttner, L. Volpe and D. Mele, "Large-scale experiments on the mechanics of pyroclastic flows: Design, engineering, and first results," *J. Geophys. Res,* p. 112, 2007.

[35] G. Dellino, P. Lino, C. Meloni, A. Rizzo, C. Bonomo, P. Giannone and S. Graziani, "Simulation-optimization in modelling ionic polymer-metal composites actuators,," *Int. J. Modell. Identif. Control, 17(1),,* pp. 8-18, 2012.

[36] Y. Morita, S. Rezaeiravesh, N. Tabatabaei, R. Vinuesa, K. Fukagata and P. Schlatter, "Applying Bayesian optimization with Gaussian process regression to computational fluid dynamics problems," *Journal of Computational Physics,* p. Volume 449, 2021.

[37] R. Gramacy, Surrogates, Gaussian Process Modeling, Design, and Optimization for the Apllied Sciences, New York: CRC Press, 2021.

[38] C. E. Rasmussen and C. K. Williams, Gaussian Process for Machine Learning, Massachusetts: MIT Press, 2006.

[39] L. Rios and N. Sahinidis, "Derivative-free optimization: A review of algorithms and comparison of software implementations," *J. Global Optim.,* pp. 1247-1293, 2013.

[40] C. Talnikar, P. Blonigan, J. Bodart and Q. Wang, "Parallel optimization for LES," *Proceedings of the Summer Program,* pp. 315-324, 2014.

[41] Y. Nabae and K. Fukagata, "Bayesian optimization of traveling wave-like wall deformation for friction drag reduction in turbulent channel flow," *Journal of Fluid Science and Technology,* p. Vol. 16 Issue 4p, 2021.

[42] J. Laurenceau, M. Meaux, M. Montagnac and P. Sagaut, "Comparison of Gradient Based and Gradient Enhanced Response Surface Based Optimizers," *AIAA Journal,* p. Vol. 48 No. 5, 2010.

[43] J. P. C. Kleijnen, Design and Analysis of Simulation Experiments, New York: Springer , 2008.

[44] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan and S. … Chintala, "PyTorch: An Imperative Style, High-Performance Deep Learning Library," *In Advances in Neural Information Processing*

Systems, vol. 32, no. Curran Associates, Inc. Retrieved, pp. 8024-8035, 2019.

[45] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel and …. others, " Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research,* vol. 12, p. 2825–2830, 2011.

[46] P. Saves, R. Lafage, N. Bartoli, Y. Diouane, J. H. Bussemaker, T. Lefebvre, J. T. Hwang, J. Morlier and J. R. R. A. Martins, "SMT 2.0: A Surrogate Modeling Toolbox with a focus on hierarchical and mixed variables Gaussian processes," *Advanced in Engineering Software,* vol. 188, p. 103571, 2024.

[47] R. Marcus and S. Blumenthal, "A Sequential Screening Procedure," *Technometrics Vol. 16,* pp. Pages 229-234, 1974.

[48] E. J. Alvarez, "Reformulated Vortex Particle Method and Meshless Large Eddy Simulation of Multirotor Aircraft," Doctoral Dissertation, Brigham Young University, Brigham , 2022.

[49] L. A. Barba, L. A. and A. C.B., "Advances in viscous vortex methods - Meshless spatial adaption based on radial basis function interpolation," *International Journal for Numerical Methods in Fluids,* vol. 47, pp. 387 - 421, 2005.

[50] J. Weber, G. Brebner and D. Kuchemann, "Low-Speed Tests on 45-deg Swept-back Wings Parts I," Aeronautical Research Council, London, 1951.

APPENDICES

A. FLOWUnsteady Unsteady Flow Solver

FLOWUnstaedy is an open-source tool which was developed by the FLOW Lab at Brigham Young University [48]. It has a capability of variable-fidelity for unsteady aerodynamics and aeroacoustics and is based on reformulated Vortex Particle Method (rVPM). rVPM is meshless CFD method solving incompressible Navier-Stokes equations with LES-filtered in their vorticity form. In addition to that, this suite also contains vortex lattice methods, strip theory blade elements, 3D panel methods and reformulated VPM. It also has capability of obtaining results for kinematic of the vehicle such as maneuvers and speed of vehicle. The software is developed in Julia and some objects are coded in C++. The suit can be compiled by both Julia and C++.

Apart of the input file with parameters and values usedused for the delta wing simulations is given below:

```
#=################################################################
# DESCRIPTION
    45° swept-back wing at an angle of attack of 4.2°. This wing has
an aspect ratio of 5.0, a RAE 101 airfoil section with 12% thickness,
and no dihedral, twist, nor taper. This test case matches the
experimental setup of Weber, J., and Brebner, G., "Low-Speed Tests
on 45-deg Swept-Back Wings, Part I," Tech. rep., 1951. The same case
is used in a VLM calculation in Bertin's Aerodynamics for Engineers,
Example 7.2, pp. 343.

# AUTHORSHIP
  * Author          : Eduardo J. Alvarez
  * Created         : Feb 2023
  * Last updated    : Feb 2023
  * License         : MIT


#=################################################################
import FLOWUnsteady as uns
import FLOWVLM as vlm

run_name    = "wing-example"  # Name of this simulation
```

```
save_path   = run_name        # Where to save this simulation
paraview    = true            # Whether to visualize with Paraview

# ----------------- SIMULATION PARAMETERS -------------------------
AOA         = 4.2                   # (deg) angle of attack
magVinf     = 49.7                  # (m/s) freestream velocity
rho         = 0.93                  # (kg/m^3) air density
qinf        = 0.5*rho*magVinf^2     # (Pa) static pressure
Vinf(X, t)  = magVinf*[cosd(AOA), 0.0, sind(AOA)]
            # Freestream function

# ----------------- GEOMETRY PARAMETERS ---------------------------
b           = 2.489     # (m) span length
ar          = 5.0       # Aspect ratio b/c_tip
tr          = 1.0       # Taper ratio c_tip/c_root
twist_root  = 0.0       # (deg) twist at root
twist_tip   = 0.0       # (deg) twist at tip
lambda      = 45.0      # (deg) sweep
gamma       = 0.0       # (deg) dihedral

# Discretization
n           = 50        # Number of spanwise elements per side
r           = 10.0      # Geometric expansion of elements
central     = false     # Whether expansion is central

# NOTE: A geometric expansion of 10 that is not central means that
the last element is 10 times wider than the first element. If
central, the middle element is 10 times wider than the peripheral
elements.

# ----------------- SOLVER PARAMETERS -----------------------------

# Time parameters
Wakelength  = 2.75*b    # (m) length of wake to be resolved
ttot        = wakelength/magVinf    # (s) total simulation time
nsteps      = 200       # Number of time steps

# VLM and VPM parameters
p_per_step  = 1         # Number of particle sheds per time step
lambda_vpm  = 2.0       # VPM core overlap
sigma_vpm_overwrite = lambda_vpm * magVinf *
            (ttot/nsteps)/p_per_step      # Smoothing core size
sigma_vlm_solver  = -1
            # VLM-on-VLM smoothing radius (deactivated with <0)
sigma_vlm_surf    = 0.05*b    # VLM-on-VPM smoothing radius
shed_starting     = true      # Whether to shed starting vortex
vlm_rlx           = 0.7       # VLM relaxation

# -----------------------------------------------------------------
```