12-11-2015

Response Surface Analysis of Trapped-Vortex Augmented Airfoils

Anup Devidas Zope

Response surface analysis of trapped-vortex augmented airfoils

By

Anup Devidas Zope

A Thesis
Submitted to the Faculty of
Mississippi State University
in Partial Fulfillment of the Requirements
for the Degree of Master of Science
in Computational Engineering
in the Bagley College of Engineering

Mississippi State, Mississippi

December 2015

Response surface analysis of trapped-vortex augmented airfoils

By

Anup Devidas Zope

Approved:

_____

J. Mark Janus
(Major Professor)

_____

Edward A. Luke
(Committee Member)

_____

David S. Thompson
(Committee Member)

_____

Pasquale Cinnella
(Graduate Coordinator)

_____

Jason M. Keith
Dean
Bagley College of Engineering

Name: Anup Devidas Zope

Date of Degree: December 11, 2015

Institution: Mississippi State University

Major Field: Computational Engineering

Major Professor: Dr. J. Mark Janus

Title of Study: Response surface analysis of trapped-vortex augmented airfoils

Pages in Study 82

Candidate for Degree of Master of Science

In this study, the effect of a passive trapped-vortex cell on lift to drag (L/D) ratio of an FFA-W3-301 airfoil is studied. The upper surface of the airfoil was modified to incorporate a cavity defined by seven parameters. The L/D ratio of the airfoil is modeled using a radial basis function metamodel. This model is used to find the optimal design parameter values that give the highest L/D. The numerical results indicate that the L/D ratio is most sensitive to the position on an airfoil's upper surface at which the cavity starts, the position of the end point of the cavity, and the vertical distance of the cavity end point relative to the airfoil surface. The L/D ratio can be improved by locating the cavity start point at the point of separation for a particular angle of attack. The optimal cavity shape (o19_aXX) is also tested for a NACA0024 airfoil.

DEDICATION

I would like to dedicate this research to my parents and my wife.

ACKNOWLEDGEMENTS

First of all, I would like to express my gratitude for my advisor Dr. Mark Janus for his continuous motivation and guidance while I conducted this research as part of my degree. Without his encouragement it would have been difficult to take this research to completion. I would also like to thank Dr. Edward Luke for his guidance regarding grid generation and use of the Loci/CHEM simulation software. His inputs were instrumental in expediting the fluid flow simulations using the Loci/CHEM software. I would also like to thank Dr. David Thompson for serving on my graduate committee. I appreciate suggestions of Jefferson Parish for his inputs on the use of radial basis functions. I thank the Center for Advanced Vehicular Systems, Mississippi State University for supporting this research.

TABLE OF CONTENTS

DEDICATION ii

ACKNOWLEDGEMENTS iii

LIST OF TABLES vi

LIST OF FIGURES vii

CHAPTER

I. INTRODUCTION 1

II. NUMERICAL SETUP 4

2.1 Grid for the FFA-W3-301 Airfoil 4
2.2 Numerical Setup 6
2.3 Grid Convergence 9
2.4 Comparison of Experimental Results and the Loci/CHEM Solution 12

III. DESIGN AND ANALYSIS OF COMPUTER EXPERIMENTS 16

3.1 Metamodeling for Computer Experiments 16
3.2 Metamodels 21
3.2.1 Polynomial Regression Models 21
3.2.2 Radial Basis Function Metamodel (RBF) (Powell, 1987) 24
3.2.3 Choice of Metamodel 25
3.3 Design of Experiments 26
3.3.1 Optimality Criteria of Experimental Designs 31
3.4 Prediction Accuracy and Regularization 33

IV. OPTIMIZATION OF THE FFA-W3-301 AIRFOIL WITH CAVITY USING METAMODELS 38

4.1 Parameterization for the Shape of Cavity on FFA-W3-301 Airfoil 38
4.1.2 Software for Design of Airfoil Profile with Cavity 41
4.2 Design of Experiment for the Cavity Parameterization 44
4.3 Software for Building Metamodels 46

4.4 Development of Metamodel of the Performance of the FFA-W3-301 Airfoil with Cavity 47
4.5 Response Surface of Lift Coefficient 49
4.6 Optimization of the Cavity Shape and Incremental Model Improvement for 12° Angle of Attack....50
4.7 The o19 Cavity at Point of Separation 56
4.8 Testing of the o19_aXX Cavity on NACA0024 Airfoil 59

V. CONCLUSION 62

REFERENCES 63

APPENDIX

A. METAMODEL DEVELOPMENT ON TOY FUNCTIONS....66

A.1 First Toy Function 67
A.2 Second Toy Function....73

B. EXPERIMENTAL DESIGN FOR THE FFA-W3-301 AIRFOIL CAVITY SHAPE OPTIMIZATION....80

LIST OF TABLES

2.1 Calculations for grid convergence test using lift coefficient as the objective function 10

2.2 Calculations for grid convergence test using drag coefficient as the objective function 10

2.3 Position of the point of separation in terms of % of chord length from leading edge of the FFA-W3-301 airfoil at various angles of attack 11

4.1 Description of parameters for the cavity design 40

4.2 Range of experimental design domain for FFA-W3-301 cavity parameters 41

4.3 Test statistics for the radial basis function metamodels developed for the lift coefficient of the FFA-W3-301 airfoil with cavity 48

4.4 Test statistics for the radial basis function metamodels developed for the drag coefficient of the FFA-W3-301 airfoil with cavity 48

4.5 The o19 cavity parameter values 51

4.6 Lift and drag coefficients for FFA-W3-301 and FFA-W3-301 with the o19 cavity for 10°, 12°, and 14° AoA 52

4.7 Lift and drag coefficients for the FFA-W3-301 airfoil and the FFA-W3-301 with the o19_a14 cavity for 14° AoA 57

4.8 Lift and drag coefficients for the NACA0024 and the NACA0024 with the o19_a12 cavity at 12° AoA 60

4.9 Lift and drag coefficients for the NACA0024 and the NACA0024 with the o19_a14 cavity at 14° AoA 60

B.1 Experimental design for training metamodels of performance of the FFA-W3-301 airfoil with cavity 81

LIST OF FIGURES

2.1 The FFA-W3-301 airfoil cross section 4

2.2 O-shaped structured grid around the FFA-W3-301 airfoil 5

2.3 Close up view of the grid around the FFA-W3-301 airfoil 6

2.4 Close up view of the grid around trailing edge of the FFA-W3-301 airfoil 6

2.5 The y+ value at each cell on the airfoil surface plotted as a function of the distance along chord from leading edge of the airfoil 7

2.6 Progress of drag and lift coefficient of the FFA-W3-301 airfoil at 12° AoA as the simulation proceeds up to 5000 iterations 8

2.7 Convergence history of mass, momentum and energy residuals for the flow over the FFA-W3-301 airfoil simulation at 12° AoA 9

2.8 Z-component of vorticity vector along airfoil surface vs distance from the leading edge for the FFA-W3-301 airfoil at various angles of attack 11

2.9 Comparison of experimental pressure coefficient with Loci/CHEM simulation pressure coefficient at various angles of attack 14

2.10 Comparison of lift and drag coefficients between the experimental data and the Loci/CHEM solution 15

4.1 The cavity design parameters 40

4.2 The MATLAB user interface of the application for generating airfoil profile with cavity 43

4.3 Structured grid in the cavity 44

4.4 Graphical user interface of application for generating SLHD and OLHD 46

4.5 Plot of response surface of the multiquadratic metamodel of lift coefficient of the FFA-W3-301 airfoil with cavity at 12° AoA. 50

4.6 The FFA-W3-301 airfoil with the o19 cavity....52

4.7 Graphs comparing pressure coefficient (Cp) and wall-shear stress magnitude between FFA-W3-301 and FFA-W3-301 with the o19 cavity at various angles of attack....53

4.8 Streamlines of flow over the FFA-W3-301 airfoil with the o19 cavity at various angles of attack 54

4.9 Graphs comparing pressure coefficient (Cp) and wall-shear stress magnitude of the FFA-W3-301 airfoil with those of the FFA-W3-301 airfoil with the o19_a14 cavity at 14° AoA 57

4.10 The FFA-W3-301 cavity with the o19_a14 cavity....58

4.11 Streamlines of flow over the FFA-W3-301 airfoil with the o19_a14 cavity at 14° AoA 58

4.12 Comparison between pressure coefficient of the NACA0024 airfoil and the NACA0024 airfoil with the o19_a14 cavity at 14° AoA....60

4.13 Profile of the NACA0024 airfoil with the o19_a12 and the o19_a14 cavities....61

A.1 Graph of function with equation no. A.1....67

A.2 Comparison of test metrics of the metamodels developed for the first toy function given by equation. A.1. 69

A.3 Graphical comparison of toy function 1 (equation A.1) and its metamodels....70

A.4 Surface plot of the second toy function given by equation A.2 74

A.5 Comparison of test metrics of the metamodels developed for the second toy function given by equation A.2....74

A.6 Plots of the metamodels developed for the second toy function given by equation A.2. 76

CHAPTER I

INTRODUCTION

Thick airfoil section wings are typically used in low speed applications such as wind turbine blades because they give higher lift at low angles of attack. Also, these airfoils give more structural strength and higher load carrying capacity. However, they suffer from the drawback of higher drag and hence lower efficiency. Thick airfoils generally experience trailing edge stall where the flow begins to separate from the trailing edge first with progressively higher angles of attack.

Vortex generators are projections mounted on external surface of an airfoil. They operate by mixing high-energy freestream flow into the boundary layer. This addition of energy into the boundary layer flow causes the fluid to stay attached longer because it has more energy to counter the adverse pressure gradient seen on the aft portion of an airfoil upper surface. Kline-Fogleman airfoils (USA Patent No. 3706430, 1970) utilize a backward facing step on the upper or lower surface (or both surfaces) of the airfoil. Such a backward facing step promotes the formation of a trapped vortex behind the step which adjusts itself with the flow and theoretically delays separation and/or provides vortex lift.

The trapped-vortex cell technique has been previously studied experimentally (Gregorio & Fraioli, 2008). The spanwise circular cavity, used in those experiments, was subjected to both passive and active flow control. The results indicated that for passive control, the vortex strength was insufficient to cause flow reattachment. The flow

separated from the starting position of the cavity. At higher angles of attack, the vortex was swept downstream by the flow, resulting in vortex shedding. The active control configuration was tested with different suction mass flow rates. The results showed that with increasing mass flow rate, the strength of the vortex inside the cavity increases and causes reattachment of the flow. If sufficient suction were applied, complete reattachment was observed.

Experimental and numerical driven cavity research has typically involved flat plate flows with an integral cavity (shear-driven cavity problem). The case here involves more than that, in that the cavity is embedded in an adverse pressure gradient. In addition to this, unlike the work done by Gregorio and Fraioli (2008), which involved experiments on only a single cavity shape, this work is focused on the goal of seeking to determine if a passive cavity shape exists which do enhances airfoil performance (ideally at different attitudes for the same fixed cavity). Initially, there was a need to identify what general geometric parameters would adequately define the configuration; then, in the event that an aerodynamic-performance improving passive design was found, what specific parameters would yield the best performance.

The current study investigated a passive trapped-vortex cell on the FFA-W3-301 wind turbine airfoil to understand its effect on performance. However, unlike the experimental study in (Gregorio & Fraioli, 2008), the cavity shape was parameterized in order to understand the effect each parameter had on the cavity flowfield and ultimately on the airfoil performance. The performance was measured using the lift and drag coefficients and the lift to drag ratio (L/D) obtained from CFD simulations. The performance metrics were modeled as a function of the cavity shape parameters using the

radial basis functions which were trained using the lift and drag data obtained at specific training points. The Loci/CHEM (Luke, Tong, Wu, Tang, & Cinnella, 2003) CFD solver was used to run simulations for various cavity configurations. The Loci/CHEM solver is a Navier-Stokes flow solver for chemically-reacting flows, developed at Mississippi State University.

CHAPTER II

NUMERICAL SETUP

2.1 Grid for the FFA-W3-301 Airfoil

The FFA-W3-301 airfoil (designed by the Aeronautical Research Institute of Sweden) is used as a Vestas wind turbine root section. The airfoil shape is as shown in figure 2.1. It has a thickness of nearly 30% of the chord and an open trailing edge of about 2% of the chord.

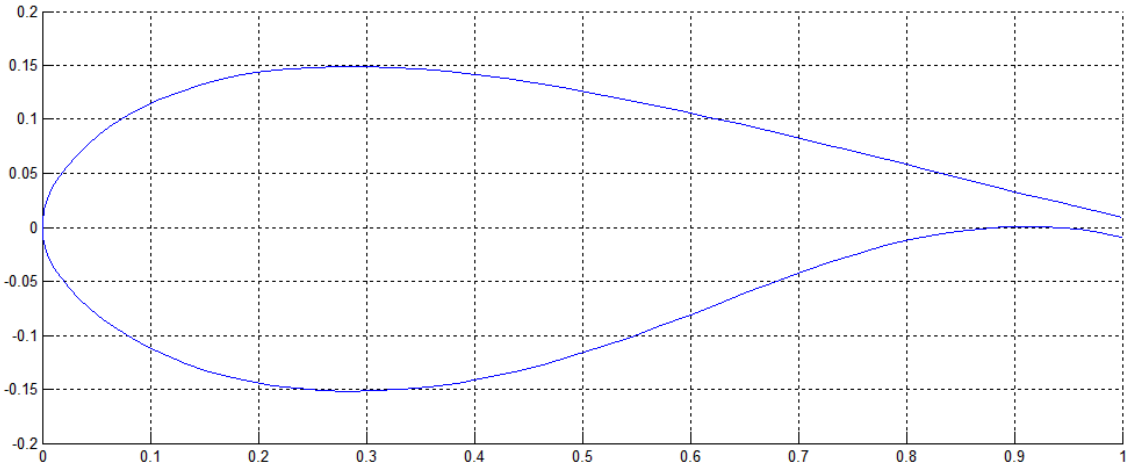


Figure 2.1 The FFA-W3-301 airfoil cross section

To generate a grid for CFD simulations, the non-zero trailing edge of the airfoil was closed with a smooth C^2-continuous cubic NURBS curve as shown in figure 2.4. A single-cell wide O-type structured grid with an elliptical outer boundary was generated

around the airfoil (figure 2.2) using the GUM-B grid generation software (Jiang & Remotigue, July 1998), developed at Mississippi State University. Because of the changing curvature of the airfoil shape profile, the flow domain had to be split into multiple blocks in order to avoid intersecting grid lines. This division also resulted into better orthogonality of the grid lines at the airfoil surface compared to the case where such division was not performed. The grid had 634750 cells in total and 2550 points on the airfoil surface. The airfoil chord length was 0.6 meters. For ease of visualization, the following figures 2.2, 2.3, and 2.4 show a coarser grid with same topology as this grid.

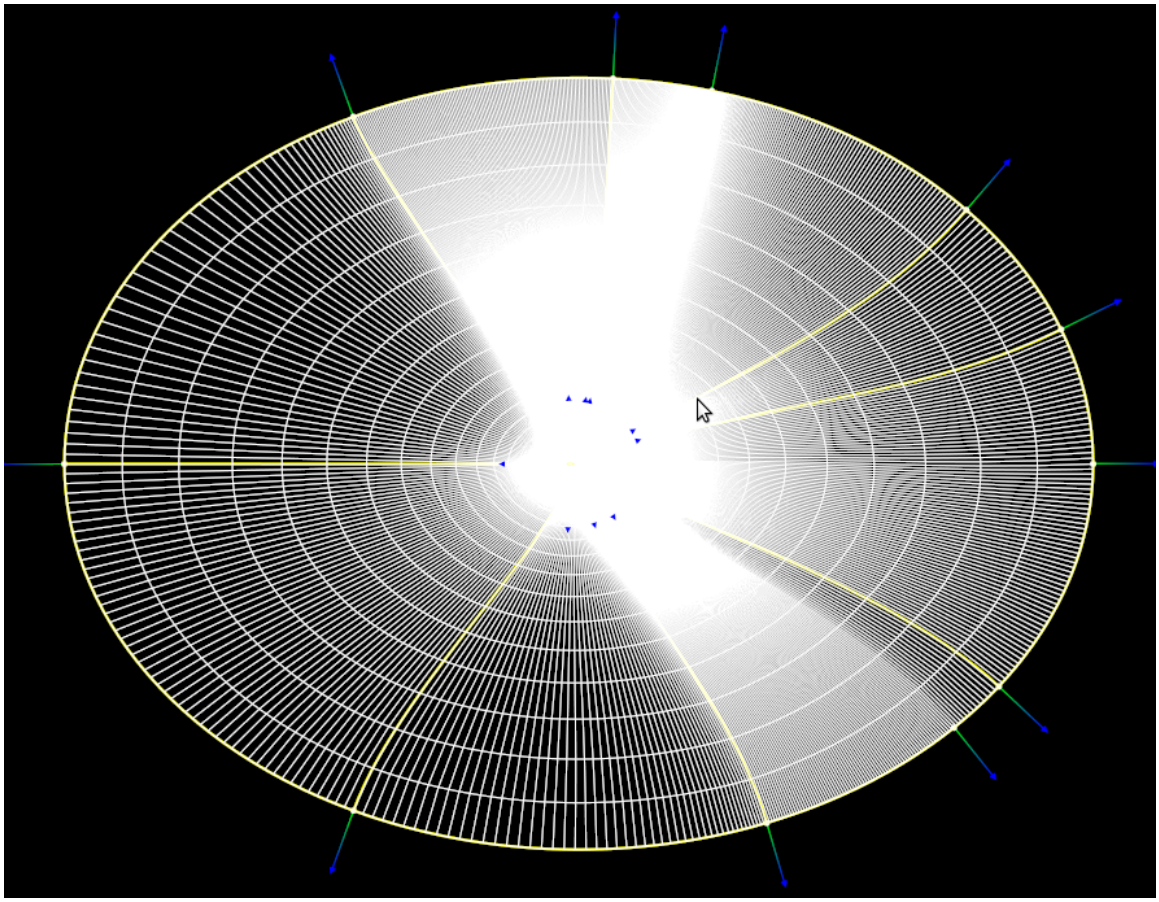


Figure 2.2 O-shaped structured grid around the FFA-W3-301 airfoil

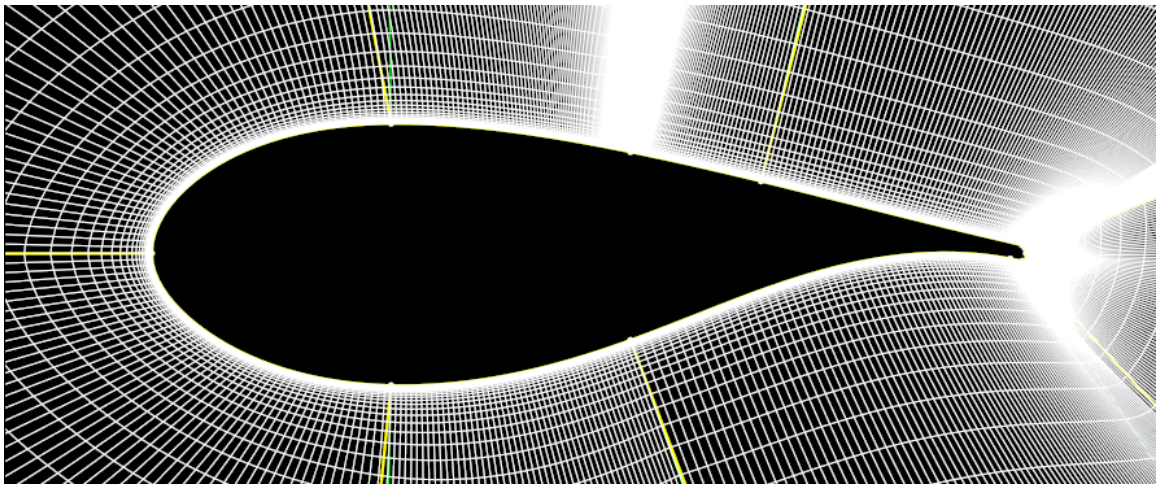


Figure 2.3 Close up view of the grid around the FFA-W3-301 airfoil

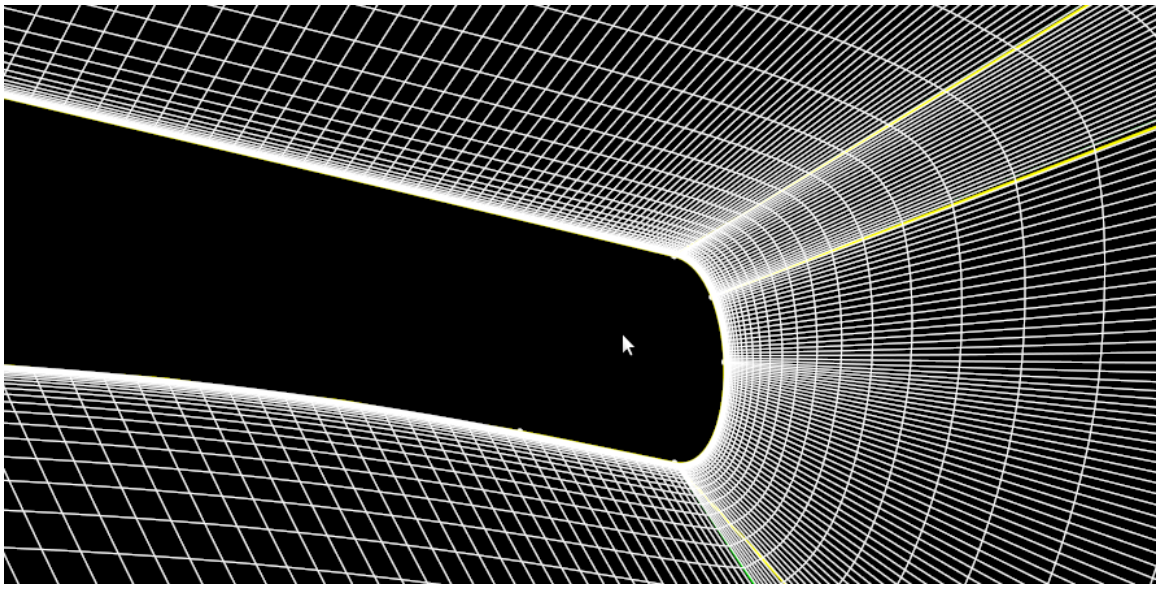


Figure 2.4 Close up view of the grid around trailing edge of the FFA-W3-301 airfoil

2.2 Numerical Setup

For the Loci/CHEM simulation of flow over the FFA-W3-301 airfoil, freestream conditions were taken as, Reynold's number $Re = 1.6 \times 10^6$, pressure $p_\infty =$

$101325\ Pa$, temperature $T_\infty = 15$ ℃. The freestream fluid was chosen as single-species air with no reactions. Since low speed flow (Mach number $M_\infty = 0.114253$) was being simulated, preconditioning was enabled. The flow over the FFA-W3-301 airfoil was solved using RANS model with Menter's shear stress transport (SST) (Menter, 1994) as the turbulence model. Time-accurate integration was performed in order to enable accurate simulations should any vortex shedding exist. Wall normal spacing was gradually reduced to 0.6×10^{-5} meters. At this wall normal spacing, the y+ graph at cells along the airfoil was as shown in figure 2.5. The y+ value was less than 1.0 at each cell on the airfoil surface indicating that the wall normal spacing was adequate to resolve turbulent boundary layer. The simulation was run for 5000 iterations during which the airfoil travels approximately 14.5 chords and the drag and lift values seemed to have converged (figure 2.6) indicating that the steady state was reached. Only the 12 degree AoA case is shown here for the y+ graph, the drag & lift coefficient graph, and the residual graph. Other angles of attack showed similar behavior.

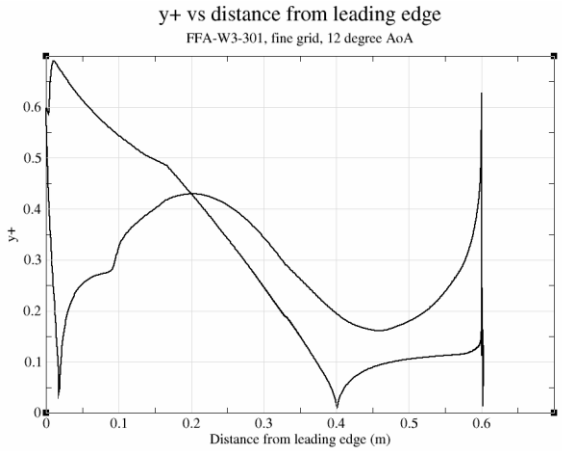


Figure 2.5 The y+ value at each cell on the airfoil surface plotted as a function of the distance along chord from leading edge of the airfoil

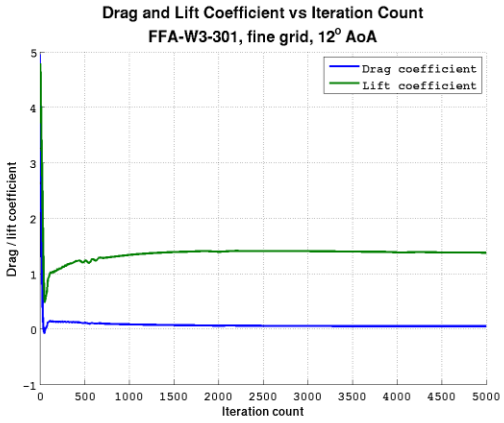


Figure 2.6 Progress of drag and lift coefficient of the FFA-W3-301 airfoil at 12° AoA as the simulation proceeds up to 5000 iterations

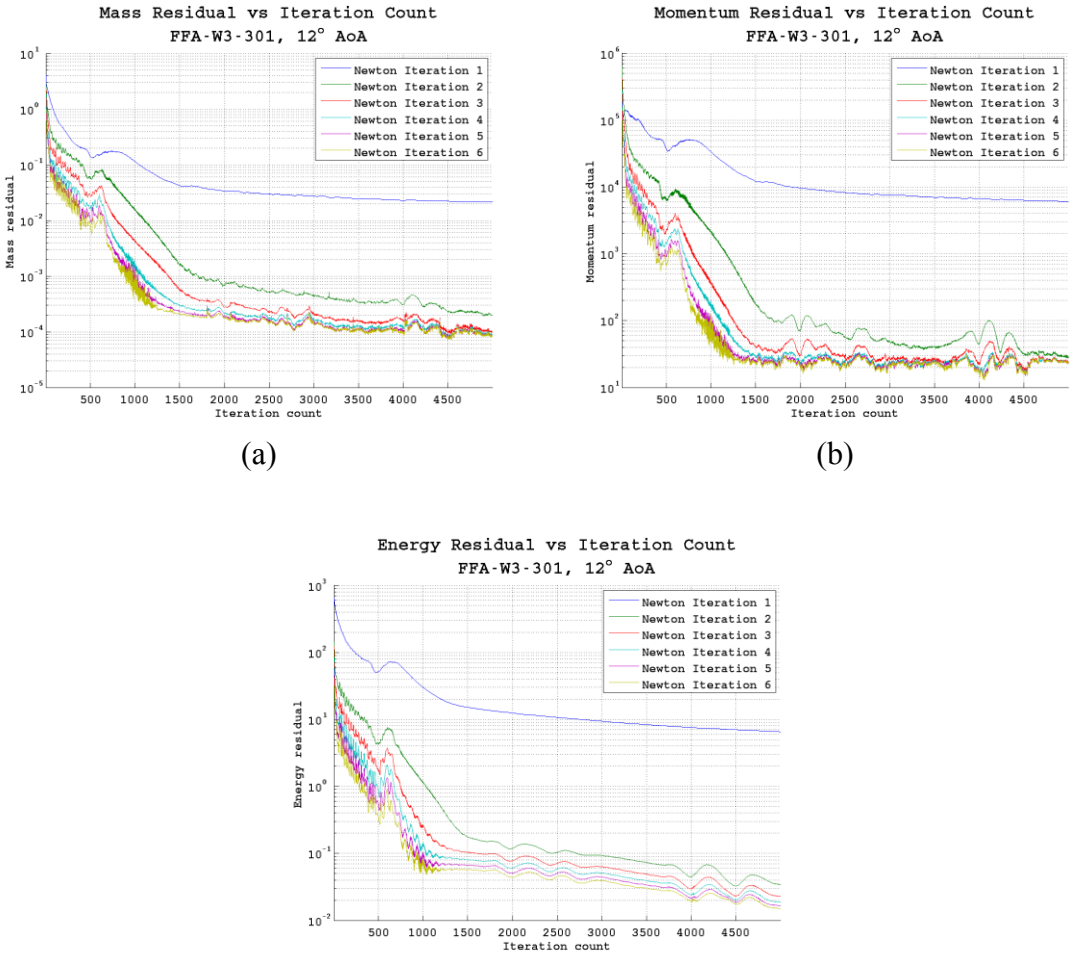


Figure 2.7 Convergence history of mass, momentum and energy residuals for the flow over the FFA-W3-301 airfoil simulation at 12° AoA

2.3 Grid Convergence

Next, a grid convergence study was performed by coarsening the original grid. Three grids were chosen - fine grid (635K cells), medium grid (157K cells), and coarse grid (39K cells) - each grid having same wall normal spacing of 0.6×10^{-5} meters but progressively coarsened spacing with grid coarsening ratio of 2.0 in each coordinate

direction between two consecutive levels. Using the procedure listed in Roache (1997), the drag and lift coefficients were checked for asymptotic convergence over the three grids for various angles of attack. The results of the grid convergence are as shown in table 2.1and table 2.2 for lift coefficient and drag coefficient respectively. The last column has values close to one indicating that the values on finer grid were in asymptotic range of convergence.

Table 2.1 Calculations for grid convergence test using lift coefficient as the objective function

AoA	Fine grid C_L (C_{L1})	Medium grid C_L (C_{L2})	Coarse grid C_L (C_{L3})	Order of conv. (p)	C_L (h=0)	GCI_{12} for C_L	GCI_{23} for C_L	Asymptotic range
2	0.4804	0.4823	0.4828	-1.9260	0.4830	-0.00671	-0.00176	1.004
4	0.7229	0.7252	0.7267	-0.6167	0.7295	-0.01143	-0.00743	1.003
6	0.9468	0.9497	0.9513	-0.8580	0.9533	-0.00854	-0.0047	1.003
8	1.1428	1.1464	1.1474	-1.8480	1.1478	-0.00545	-0.00151	1.003
10	1.2951	1.3001	1.3025	-1.0589	1.3047	-0.00928	-0.00444	1.004
12	1.3760	1.3855	1.3892	-1.3604	1.3916	-0.01414	-0.00547	1.007

Table 2.2 Calculations for grid convergence test using drag coefficient as the objective function

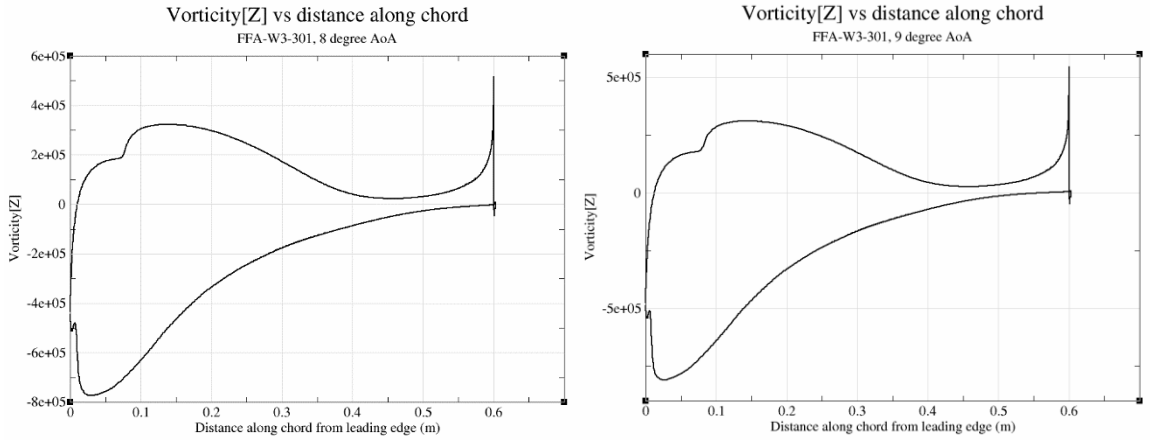
AoA	Fine grid C_D (C_{D1})	Medium grid C_D (C_{D2})	Coarse grid C_D (C_{D3})	Order of conv. (p)	C_D (h=0)	GCI_{12} for C_D	GCI_{23} for C_D	Asymptotic range
2	0.0191	0.0195	0.0219	2.5850	0.01902	0.005236	0.030769	1.0209
4	0.0217	0.0221	0.0249	2.8073	0.02163	0.00384	0.026395	1.0184
6	0.0256	0.0261	0.0295	2.7655	0.02551	0.004209	0.028075	1.0195
8	0.0309	0.0315	0.0356	2.7726	0.03080	0.004161	0.027891	1.0194
10	0.0378	0.0385	0.0435	2.8365	0.03769	0.003768	0.026427	1.0185
12	0.0487	0.0497	0.0557	2.5849	0.0485	0.005133	0.030181	1.0205

The point of separation for various angles of attack was determined using the z-component of the vorticity vector (figure 2.8). For an angle of attack of 8° or less, there is no separation. Separation starts somewhere between 8° and 9° angle of attack. The distance of the separation point from the airfoil's leading edge in terms of percentage of chord length is listed in table. 2.3 for various angles of attack.

Table 2.3 Position of the point of separation in terms of % of chord length from leading edge of the FFA-W3-301 airfoil at various angles of attack

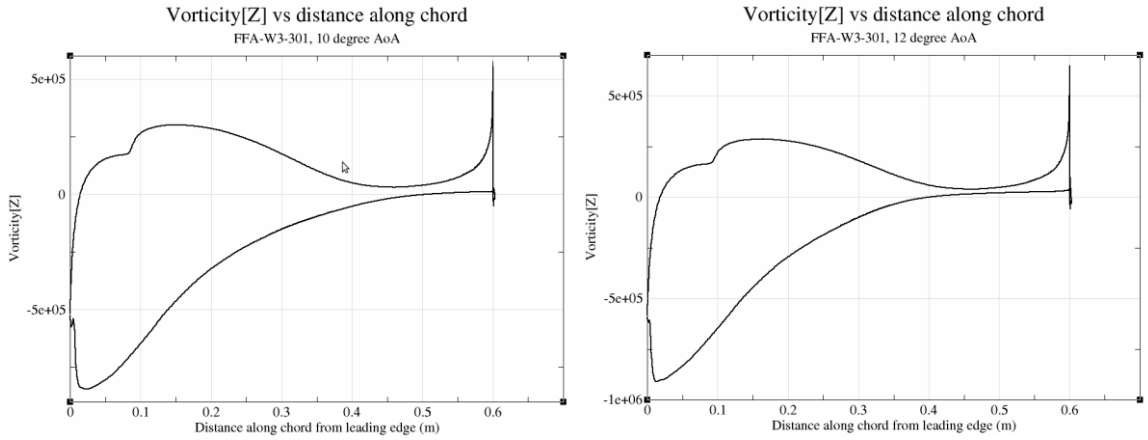
Angle of attack	**% chord distance of the point of separation**
8	No separation
9	91.6%
10	83.3%
12	66.6%
14	50.0%

The % distance to the separation point is approximate



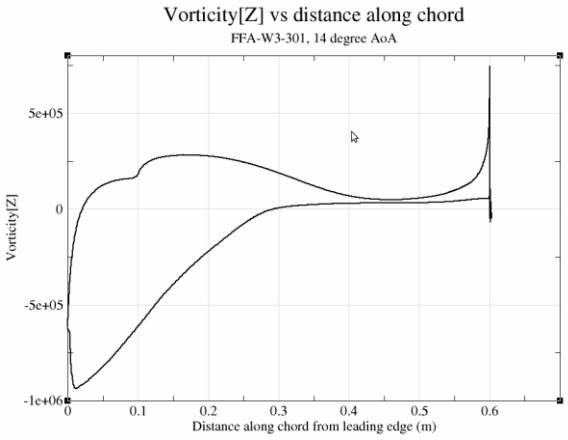
(a) (b)

Figure 2.8 Z-component of vorticity vector along airfoil surface vs distance from the leading edge for the FFA-W3-301 airfoil at various angles of attack



(c)

(d)



(e)

Figure 2.8 (continued)

Since the flow simulation is 2D, vorticity vector has only the Z-component

2.4 Comparison of Experimental Results and the Loci/CHEM Solution

(Fuglsang, Antoniou, Dahl, & Madsen, 1998) contains results of wind tunnel tests on the FFA-W3-301 airfoil. A comparison of the pressure coefficient values between the Loci/CHEM simulation and the experiments is presented in figure 2.9. These graphs show a very close comparison of the Loci/CHEM pressure coefficient with the experimental pressure coefficient up to 8° angle of attack. The agreement with the

experimental data decreased at higher angles of attack. The lift and drag coefficient comparison is presented in figure 2.10. The Loci/CHEM solution gave higher lift and lower drag compared to the experiments for angles of attack more than 10°. This is due to Loci/CHEM predicting a lower suction peak on the airfoil. A comparison of the point of separation at various angles of attack is also necessary. However, the point of separation for various angles of attack was not reported by Fuglsang et al. (1998); however, they mentioned that separation began to occur at around 11.2° angle of attack. In the Loci/CHEM solutions, separation began at around 9° angle of attack. These numerical results, though not in exact agreement with the experimental results for separated flows, were consistent with the experimental results.

Moreover, no experimental data was available for the FFA-W3-301 airfoil with cavity. Hence, in this study, the metamodel of the performance of the FFA-W3-301 airfoil with a cavity was built using the numerical results and was compared with the numerical results on the unmodified FFA-W3-301 airfoil in order to understand the effects of the cavity shape on the airfoil's performance. Therefore the discrepancy would not affect the conclusions of this study. In order to perform this numerical study, metamodels that relate cavity shape with the airfoil's lift and drag coefficient are developed. Next chapter discusses the metamodel development process.

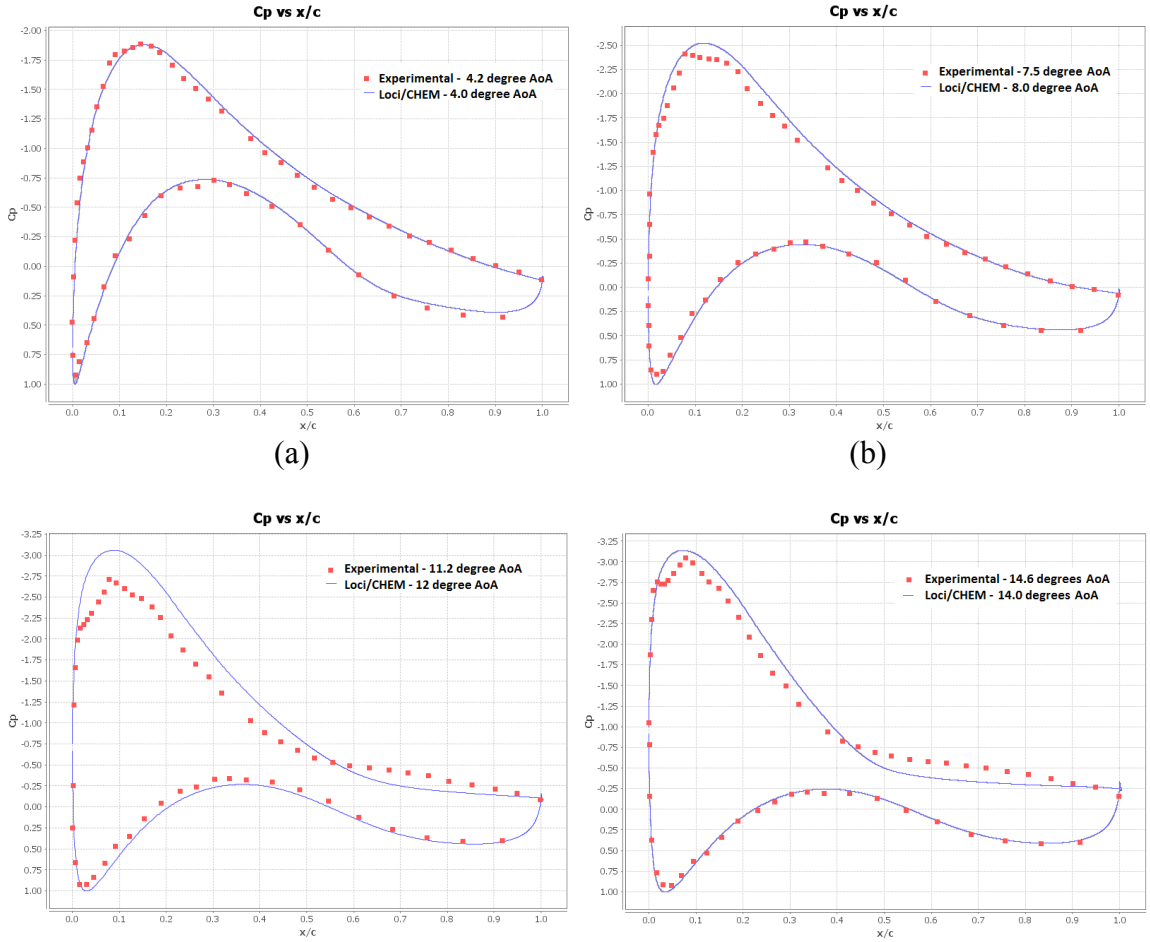
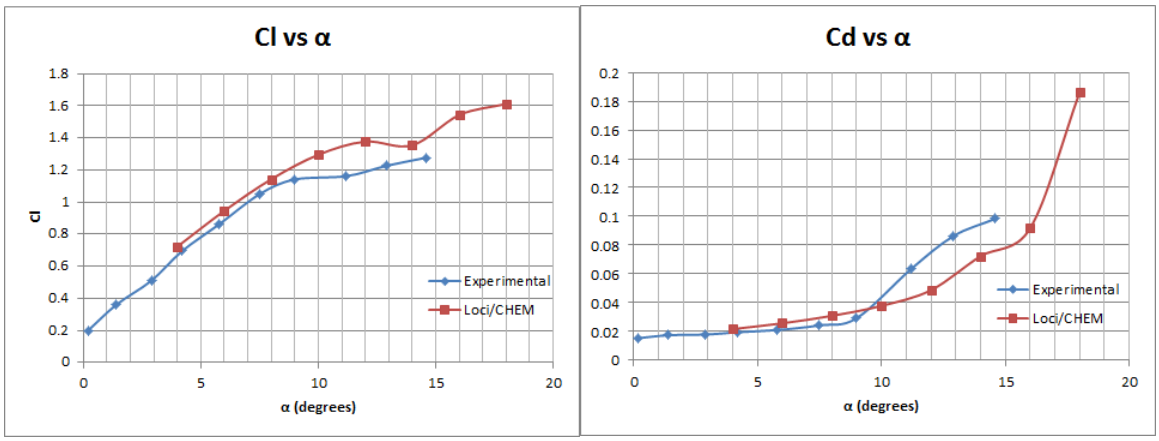
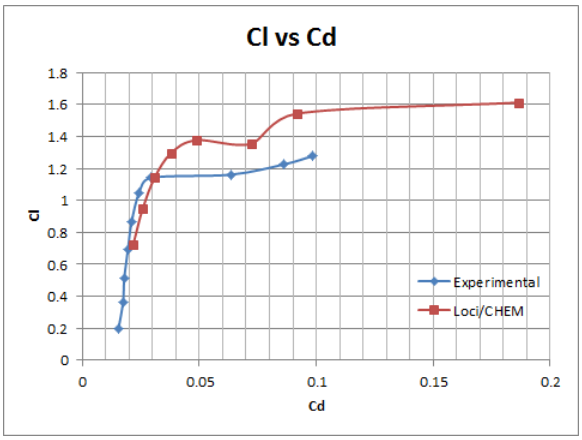


Figure 2.9 Comparison of experimental pressure coefficient with Loci/CHEM simulation pressure coefficient at various angles of attack



(a) (b)



(c)

Figure 2.10 Comparison of lift and drag coefficients between the experimental data and the Loci/CHEM solution

CHAPTER III

DESIGN AND ANALYSIS OF COMPUTER EXPERIMENTS

3.1 Metamodeling for Computer Experiments

Though carefully conducted physical experimentation is a reliable way to understand a scientific phenomenon, conducting such experiments may be time consuming and expensive. In some cases, it is impossible to conduct such experiments. Advancements in computer architectures and numerical modeling have enabled investigation of scientific phenomena with ease, accuracy and speed. Hence, computer simulations are now being used for such purposes. For example, CFD simulation can be performed to determine an airfoil's lift and drag coefficient with a fairly good accuracy without the need to perform wind tunnel tests. However, some numerical schemes used to simulate fluid flow are computationally so expensive that it takes several hours to determine lift and drag coefficients of a single airfoil design. If aim is to find out an airfoil design that maximizes lift, then the long time required for simulations makes it difficult to explore all possible airfoil designs and choose most optimum airfoil shape. Therefore, a computationally cheap model is required that replaces the original expensive simulation, which can be used for evaluation of lift and drag coefficients with reasonable accuracy. This substitute model is called a metamodel or a surrogate model. It approximates a process for which a simple functional relationship between its inputs and

outputs is not known. Exploration of inputs values to the process that yield most optimum output becomes faster and efficient with a metamodel.

A process, $\mathcal{P}$, can be thought as a black box that takes one or more inputs and produces one or more outputs. However, in this thesis, only one response is considered for a process. An input parameter to a process is also called an input variable or a factor. An output from a process is also called a response or an effect. The process can be expressed in function form as $y = \mathcal{P}(\vec{x})$, where y is the response corresponding to the values of the inputs specified by the vector $\vec{x}$. The metamodel development for a process involves the following steps.

1) Parameterization: This step involves choosing the input variables, let's say x_1 to x_s, that can possibly have impact on the output of the process, $\mathcal{P}$. It is also necessary to choose the range of values for each variable such that it covers the space of exploration called the design space, denoted by $\mathcal{D}$. Parameterization and domain selection require some prior knowledge of the process or some experimentation.
2) Choosing a metamodel: This is an important step because different metamodels perform differently in terms of training complexity, computational complexity and prediction accuracy as well as specific requirements for sampling. A metamodel is denoted by $\mathcal{M}$ and the response from the metamodel is denoted by $\hat{y} = \mathcal{M}(\vec{x})$ corresponding to the vector of input variables $\vec{x}$.
3) Design of Experiment: This step is concerned with choosing points, called training points, from the design space, $\mathcal{D}$, such that the design helps develop

as accurate model as possible with smallest sample size. The chosen points are also called design points, denoted by $d = \{\vec{x}_i \mid 1 \leq i \leq n, \vec{x}_i \in \mathcal{D}\} = [\vec{x}_1, \vec{x}_2, \dots, \vec{x}_n]^T$. Each $\vec{x}_i$ is a s-dimensional vector. Therefore, d is a matrix of dimensions $n \times s$.

4) Generation of training data set: After an experimental design is obtained, actual experiments are conducted for the level set combinations of each point in the experimental design and the corresponding response is obtained, denoted by $y_S = \{y_i \mid y_i = \mathcal{P}(\vec{x}_i), 1 \leq i \leq n\} = [y_1, y_2, \dots, y_n]^T$. The design points along with the obtained responses are called the training data set, denoted by $\mathcal{T} = \{(y_i, \vec{x}_i) \mid 1 \leq i \leq n\}$.
5) Model building: The next step is to train the chosen metamodel using the training data set, $\mathcal{T}$. It involves finding values of constants in the chosen metamodel.
6) Model validation: The final step is to assess the usefulness and accuracy of the trained metamodel using various criteria. If the model is not accurate enough, then a different metamodel can be selected and/or more training data points can be added to the experimental design.

For a physical experiment, a change in its outputs can occur because of change in one or more inputs or because of random errors introduced while performing experiments. In a case like this, statistical model building techniques need to distinguish the effect due to random errors from the effect due to change in input variable values. The ANOVA (analysis of variance) framework with data collected at points of a factorial design is a commonly used methodology for conducting this type of analysis for physical

experiments (Box & Draper, 2007). It relies on the techniques such as randomization, blocking and replication to yield meaningful interpretations about the coefficients of a polynomial model used for approximating the physical process. Unlike a physical process, a computer code does not have any source of random errors, that is, for a given set of input parameter values, it always produces same output values. Therefore the techniques such as randomization, replication and blocking and the ANOVA analysis is irrelevant. The inputs to a deterministic computer experiment can be perturbed with random errors through the use of random number generators, and the statistical analysis can be conducted for such a setup to get meaningful interpretations (Kleijnen, 1986). However, (Sacks, Welch, Mitchell, & Wynn, 1989) pointed out that since computer experiments inherently don't have a source of random errors, a different set of analysis and model building techniques are more suitable for their analysis. They further showed that even though computer experiments are deterministic, their model building can be thought as a stochastic process and present a statistical framework for the design and analysis of computer experiments based on the Kriging model (Matheron, 1963). Sacks et al. stated that a computer experiment can be represented by the model in equation 3.1. It has a polynomial model that forms the deterministic part of the formulation. It does not give accurate response value. Therefore, the model has an error term $Z(\vec{x}) \sim N(0, \sigma^2)$ that models the deviation of the observed response from the deterministic part.

$$Y(\vec{x}) = \sum_{j=1}^{p} \beta_j B_j(\vec{x}) + Z(\vec{x}) \tag{3.1}$$

This makes the function $Y(\vec{x})$ a stochastic process as well. Then, considering the observed data $\vec{y}_d = [y(\vec{x}_1), \dots, y(\vec{x}_n)]^T$ at sites $d = [\vec{x}_1, \dots, \vec{x}_n]^T$, which can be thought as a vector of random variables Y_d, the equation 3.2 was obtained for the predictor

$\hat{y}(\vec{x}) = c^T Y_d$ at an untried site $\vec{x}$, where $c(\vec{x})$ is a vector of weights. This was done by minimizing the mean square error, $MSE[y(\vec{x})] = E[(c^T Y_d - Y(\vec{x}))^2]$, of the predictor subjected to the constraint of unbiasedness, that is, $E[c^T Y_d] = E[Y(\vec{x})]$.

$$\hat{y}(\vec{x}) = \sum_{j=1}^{p} \hat{\beta}_j f_j(\vec{x}) + r^T R^{-1}\left(Y_d - \sum_{j=1}^{p} \hat{\beta}_j f(x)\right) \quad (3.2)$$

Here, $\hat{\beta}_j$are the generalized least square estimates of the coefficients of the polynomial model, $R = [R(\vec{x}_i, \vec{x}_j)]$ is the matrix of correlations of errors at the design sites, and $r = R(\vec{x}_i, \vec{x})$ is the vector of correlation between the errors at the design sites and the untried site. Thus, the fitted model for a computer experiment is a combination of deterministic polynomial model that captures global trends of the response and a random process that interpolates the residuals for local correction at the untried site $\vec{x}$. The choice of correlation function $R(\vec{v}, \vec{w})$ depends on the nature of the responses collected at the design sites. Sacks et al. chose $R(\vec{v}, \vec{w})$ from a product form of family of functions. They further stated that selecting the design for computer experiments is a statistical problem of choosing the design points according to a correlation function. Kriging is a type of metamodel used for computer experiments. There are a number of other metamodels that can be used such as radial basis functions (Powell, 1987), neural networks (Lawrence, 1994), multivariate adaptive regression splines (MARS) (Friedman, 1991), etc.

On a broad level, a metamodel falls in one of the two categories – a parametric model or a non-parametric model. A parametric model has a fixed functional form. A non-parametric model, on the other hand, doesn't have any such form. Instead, it has a function for each data point in the training data set and the final form of the metamodel is a combination of these functions. Both types of models have constants that can be tuned

to a given data set. The process of training a model involves finding values of these constants such that the resulting metamodel fits the training data in the best possible way.

Most of the metamodels used in practice are linear with the form as in equation 3.3.

$$y = \sum_{i=0}^{p} \beta_i B_i(\vec{x}) \tag{3.3}$$

Here, B_i are basis functions and β_i are coefficients or weights. For computer codes, many types of models are proposed in literature (Fang, Li, & Sudjianto, 2006) (Mullur & Messac, 2005) (Friedman, 1991) (Lawrence, 1994) (Powell, 1987), some of which are discussed below.

3.2 Metamodels

3.2.1 Polynomial Regression Models

A linear polynomial model of degree one with single output and s inputs is as shown by equation 3.4. For s input variables, this model has $p = s + 1$ unknowns.

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_s x_s \tag{3.4}$$

Such a linear polynomial model of degree one is of limited use if the relationship between inputs and output is nonlinear. Hence, higher degree polynomial models are required. If second-degree bilinear interaction effects are included in the above polynomial model, then it has the form shown in equation 3.5. This model has $p = s(s + 1)/2 + 1$ unknowns.

$$y = \beta_0 + \beta_1 x_1 + \cdots + \beta_s x_s + \beta_{s+1} x_1 x_2 + \cdots + \beta_{s(s+1)/2} x_{s-1} x_s \tag{3.5}$$

If quadratic terms are also included, then the model takes the form as shown in following equation. This model has $p = (s + 1)(s + 2)/2$ terms.

$$y = \beta_0 + \beta_1 x_1 + \cdots + \beta_{s+1} x_1 x_2 + \cdots + \beta_{\frac{s(s+1)}{2}+1} {x_1}^2 + \cdots + \beta_{\frac{s(s+3)}{2}} {x_s}^2 \qquad (3.6)$$

In general, a polynomial model can be expressed in a form as shown in equation 3.3. For example, in the case of polynomial model with bilinear interaction effects, the basis functions are $B_0 = 1, B_1 = x_1, \dots, B_p = x_{s-1}x_s$. The unknown coefficients in the polynomial models can be determined by substituting the training data set $\mathcal{T} = \{(y_i, \vec{x}_i) \mid 1 \leq i \leq n\}$ in the chosen polynomial model equation. This gives a system of equations as shown in equation 3.7.

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} B_0(\vec{x}_1) & B_1(\vec{x}_1) & \dots & B_p(\vec{x}_1) \\ B_0(\vec{x}_2) & B_1(\vec{x}_2) & \dots & B_p(\vec{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ B_0(\vec{x}_n) & B_1(\vec{x}_n) & & B_p(\vec{x}_n) \end{bmatrix} \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_p \end{bmatrix} \rightarrow \boldsymbol{y} = \boldsymbol{B\beta} \qquad (3.7)$$

If $n = p$, the solution can be obtained accurately using $\boldsymbol{\beta} = \boldsymbol{B^{-1}y}$ as long as $\boldsymbol{B}$ is far from singular. If $n > p$, the system of equations is overdetermined. A unique solution can be obtained if some constraint is imposed on their values. For this, a cost function $C(\boldsymbol{\beta})$ is considered. Minimization of the cost function gives unique coefficients denoted by $\hat{\beta}_i$ since they give a polynomial model that approximates the relationship between the inputs and the output. If the cost function is square of the L_2-norm of residual, that is, $C(\boldsymbol{\beta}) = \|\boldsymbol{y} - \boldsymbol{B\beta}\|^2$, the minimization problem can be stated as shown in equation 3.8.

$$\widehat{\boldsymbol{\beta}} = \min_{\boldsymbol{\beta} \in R^p} \|\boldsymbol{y} - \boldsymbol{B\beta}\|^2 \qquad (3.8)$$

This can be solved by taking the derivative of the cost function with respect to each β_i and equating each equation to zero. This gives p equations in p unknowns, which can be solved simultaneously to obtain values of $\hat{\beta}_i$. This is an ordinary least square (OLS) estimation of the coefficients given by equation 3.9. Since the cost function in this case is

quadratic and concave upwards, the OLS solution is guaranteed to be at the minima of the cost function.

$$\boldsymbol{\beta} = (\boldsymbol{B}^T\boldsymbol{B})^{-1}\boldsymbol{B}^T\boldsymbol{y} \tag{3.9}$$

The drawback of the OLS estimator is that if $\boldsymbol{B}^T\boldsymbol{B}$ is far from nearly orthogonal, then the OLS estimators are sensitive to errors and tend to give large coefficients. This makes the polynomial model useless. One approach to deal with this ill-conditioning is to introduce additional information into the equations in the form of a penalty for the coefficients. In case of ridge regression (Tikhonov & Arsenin, 1978), the penalty is the L_2-norm of the coefficient vector. Then the minimization problem can be stated in Lagrange multiplier form as shown in equation 3.10.

$$\widehat{\boldsymbol{\beta}} = \min_{\boldsymbol{\beta}\in R^p}\left(\frac{1}{2}\|\boldsymbol{y} - \boldsymbol{B}\boldsymbol{\beta}\|^2 + \lambda\|\boldsymbol{\beta}\|^2\right) \tag{3.10}$$

The solution to this minimization problem is given by equation 3.11.

$$\beta = (\boldsymbol{B}^T\boldsymbol{B} + 2\lambda\boldsymbol{I})^{-1}\boldsymbol{B}^T\boldsymbol{y} \tag{3.11}$$

The penalty λ applied to the L_2-norm of the coefficients tends to make a compromise between expectation and variance (Hoerl & Kennard, 1970). The choice of λ affects the estimators and the prediction accuracy of the generated model. The choice of λ can be made using a procedure called regularization.

If the penalty function is the L_1-norm of the coefficient vector, then the regression is called LASSO (Least Absolute Shrinkage and Selection Operator), first proposed by Tibshirani (1996). In this case, the minimization problem is harder to solve analytically as the penalty function is not differentiable at zero. Several algorithms are proposed in (Tibshirani, 1996) to get a solution to the LASSO estimators. The L_1-norm penalty

function tends to make some of the coefficients zero indicating that the associated basis functions have a negligible effect on the response and can be neglected. This is similar to the subset selection. LASSO combines the benefits of both subset selection and ridge regression.

3.2.2 Radial Basis Function Metamodel (RBF) (Powell, 1987)

The radial basis function (RBF) (Powell, 1987) metamodel is a scalar-valued function formed by linear combination of basis functions called radial functions. Each radial function is a function of the Euclidian distance of a point from a certain fixed point in space called the center of the radial function. Hence the radial function have the same value at all points equidistant from the center of the function. RBFs can be easily extended to higher dimensions enabling easy modeling of multivariate functions. The form of a radial basis functions metamodel is as shown in equation 3.12.

$$y(\vec{x}) = \sum_{k=1}^{n} \beta_k \varphi(\|\vec{x} - \vec{x}_k\|) \tag{3.12}$$

Here, β_k is scalar coefficient of the k[th] radial basis function, φ is chosen radial function, $\vec{x}_k$ is center of the k[th] radial function, $\|\quad\|$ is Euclidian norm operator. There are many forms of radial basis function kernels as mentioned below:

Gaussian radial function

$$\varphi(\|\vec{x} - \vec{x}_k\|) = \exp\left(-\frac{\|\vec{x}-\vec{x}_k\|}{r^2}\right) \tag{3.13}$$

Multiquadratic radial function

$$\varphi(\|\vec{x} - \vec{x}_k\|) = \frac{\sqrt{r^2+\|\vec{x}-\vec{x}_k\|}}{r} \tag{3.14}$$

Inverse multiquadratic radial function

$$\varphi(\|\vec{x} - \vec{x}_k\|) = \frac{r}{\sqrt{r^2+\|\vec{x}-\vec{x}_k\|}} \tag{3.15}$$

Inverse quadratic radial function

$$\varphi(\|\vec{x} - \vec{x}_k\|) = \frac{r^2}{r^2 + \|\vec{x} - \vec{x}_k\|} \tag{3.16}$$

Here, r is attenuation factor. The choice of r affects the model's prediction accuracy and can be chosen using regularization.

3.2.3 Choice of Metamodel

Jin, Chen and Simpson (2000) presented a systematic study about the performance of various types of metamodels trained using different types of sample sizes for a number of benchmark problems with different characteristics, trained using sparse, small and large training data set. The types of metamodels considered in the paper were polynomial regression (PR), radial basis functions (RBF), Kriging (KG) and multivariate adaptive regression splines (MARS). The accuracy of the trained metamodels was measured using mean square error (MSE), relative maximum average error (RMAE), and the relative average absolute error (RAAE) using a sufficiently large validation data set. They categorized the problem based on the number of variables into large scale (more than 10 variables) and small scale (2 or 3 variables). The second type of categorization was based on the R^2 value of the second-order polynomial model for the problem. A problem was considered high-order, non-linear if $R^2 \geq 0.99$ and low-order non-linear otherwise. Third type of categorization was smooth behavior or noisy behavior depending on whether the response has artificially introduced random error or not. The training data set was either a scarce set ($3s$ training points), a small set ($10s$ training points), or a large set ($3(s+1)(s+2)/2$ training points). The accuracy of a metamodeling technique was measured using the mean of each of the performance metrics mentioned above, while its

robustness was measured using the variance of the accuracy metric. They found that, overall, RBF and KG performed better than the other models for all of the test problems and any sample size in terms of their accuracy and robustness. The accuracy of the PR model decreased significantly as the non-linearity of the problems increased while the robustness of PR, RBF and KG decreased as the non-linearity of the problems increased. The RBF performed the best under different scales of the problems. Accuracy of MARS dropped significantly as sample size decreased. KG lost its accuracy for a noisy response. Thus, in their study, the RBF model performed best for high-order, non-linear problems with any scale and any sample size. Therefore for the purpose of this study, radial basis function metamodels were used.

3.3 Design of Experiments

The training data is collected by conducting experiments at carefully chosen points in the design space. The process of choosing training data points is called designing an experiment. The response surface (polynomial regression) models are traditionally trained using factorial designs (Box & Draper, 2007). A factorial design for s variables with a levels for each variable is denoted by a^s. If we imagine an s-dimensional space, then a is the number of uniformly-spaced points in a variable's range along its coordinate axis of the hyperspace. For example, a three-level, full factorial design in two variables, x_1 and x_2, is denoted by 3^2. It has total $3^2 = 9$ training points.This design can be used to train a polynomial model with linear, interaction and quadratic effects. With a two-level factorial design, only linear and interaction effects can be modeled. Thus, as the degree of polynomial model increases, the number of training points in the factorial design needs to increase exponentially (a^2 in case of two

variables).This dramatic increase in the number of training points of a factorial design makes them unattractive when either the degree of the polynomial is high or when the number of variables is large.

The Latin hypercube design (LHD) was introduced in (McKay, Beckman, & Conover, 2000). It is an effective method for designing experiments for computer codes. McKay, Beckman and Conover reported that a LHD produces the least sample variance compared to random and stratified sampling. A random LHD was constructed by dividing the range of each variable in a number of strata and choosing one training point randomly from each stratum with equal probability. Thus, the number of training points is equal to the number of strata. Unlike factorial design, in a LHD, the number of training points is independent of the number of variables. A LHD with n training points and s input variables is denoted by $LHD(n, s)$. It can be stored in a $(n \times s)$ matrix with each row representing a design point. If instead of choosing randomly from each stratum, the design points are chosen at the mid-point of each stratum, then it is called mid-point Latin hypercube design, $MLHD(n, s)$. The LHD generated this way has uniform distribution along each individual variable dimension. However, it may not necessarily have a uniform distribution over higher dimensions. A model trained using such a design may perform poorly in terms of prediction accuracy in the regions of the hyperspace where there are fewer training points. Therefore, good space-filling property is essential for a LHD. There are $n!^2$ possible $MLHD(n, s)$ designs and searching the best space-filling design is NP-hard problem. Therefore, the best LHD design search can be restricted to a special class of LHDs with specific properties. Two such designs are orthogonal column Latin hypercube design (OLHD) and symmetric Latin hypercube design (SLHD). An

OLHD has zero correlation among the columns of the design matrix, that is, the dot product of any two columns is zero. This is useful for building a polynomial model with bilinear interaction effects because the coefficients of the bilinear interaction basis functions in the model are not correlated with the coefficients of the linear basis functions. (Ye, 1998) gave an algebraic algorithm to generate OLHD when the number of training points is a power of 2 or power of 2 plus 1. The procedure is as outlined below. For a given $m \geq 2$, this algorithm generates OLHD with $n = 2^m$ or $n = 2^m + 1$ training points for $2m - 2$ variables. The algorithm constructs the top half (let's say $\boldsymbol{T}$) of the OLHD and then reflects it along the center point to produce the bottom half of the design matrix. A magnitude matrix (let's say $\boldsymbol{M}$) of the top half is constructed by producing permutations of $e = [1,2, \dots, 2^{m-1}]^T$ as follows.

$$\boldsymbol{M} = \left[e, \boldsymbol{A}_i e, \boldsymbol{A}_{m-1} \boldsymbol{A}_j e\right]; i = 1, \dots, m-1; j = 1, \dots, m-2 \tag{3.17}$$

Where matrix $\boldsymbol{A}_k$ is given by following formula.

$$\boldsymbol{A}_k = \underbrace{\boldsymbol{I} \otimes \dots \otimes \boldsymbol{I}}_{m-1-k} \otimes \underbrace{\boldsymbol{R} \otimes \dots \otimes \boldsymbol{R}}_{k} \tag{3.18}$$

Here, $\boldsymbol{I} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ and $\boldsymbol{R} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$.

A sign matrix (let's say $\boldsymbol{S}$) is generated as follows.

$$S = \left[\mathbf{1}, a_i, a_1 a_{j+1}\right]; i = 1, \dots, m-1; j = 1, \dots, m-1 \tag{3.19}$$

Here,

$$a_k = \boldsymbol{B}_1 \otimes \boldsymbol{B}_2 \otimes \dots \otimes \boldsymbol{B}_k \otimes \dots \otimes \boldsymbol{B}_{m-1}; k = 1, \dots, m-1 \tag{3.20}$$

Where, $\boldsymbol{B}_k = a = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$ and $\boldsymbol{B}_i = \mathbf{1} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, i \neq k$

The top half of the OLHD is then given by the element-wise product of the magnitude and the sign matrix, hence $\boldsymbol{T} = \boldsymbol{M} \circ \boldsymbol{S}$. An OLHD with 2^m design points is generated by removing the midpoint of the $2^m + 1$ design and rescaling. The complete design matrix is obtained by reflecting the top part. However, limitations of this algorithm include applicability for even number of variables, fixed number of design points for a given the number of variables, and no uniformity of the points in the design space. The space-filling quality of the OLHD can be improved by generating a number of OLHD's using different starting permutation vectors e and choosing the design that evaluates as being the best according to the selected optimality criteria (Ye, 1998).

Other approaches for generating an OLHD are also mentioned in the literature. Cioppa & Lucas (2007) extended the approach of Ye (1998) to generate OLHD with same number of design points but for a larger number of variables. The magnitude matrix in their algorithm is given by,

$$\boldsymbol{M} = \left[e, \boldsymbol{A}_i e, \boldsymbol{A}_j \boldsymbol{A}_{j+1}\right]; i = 1, \dots, m-1; j = 1, \dots, m-2. \tag{3.21}$$

This strategy produces a design with $m + \binom{m-1}{m}$ columns which are orthogonal to each other. Further, Cioppa & Lucas (2007)showed that such a design does not have good space-filling property and suggested sacrificing strict orthogonality, producing the nearly orthogonal Latin hypercube designs (NLHD). They also proposed deleting a few columns from the design matrix if the number of variables is not exactly the same as $m + \binom{m-1}{m}$ and then inserting new design points to ensure a good space-filling property as well as maintaining near orthogonality. The method proposed by (Steinberg & Lin, 2006)

consists of rotating a 2^k factorial design, when k is power of two ($k = 2^m$) and 2^k maximum number of factors.

These OLHDs suffer from the drawback that the number of runs are fixed, based on the number of variables. They also exhibit a poor space-filling property. Morris and Mitchell (1995) found that many optimal LHDs possess the symmetry property. They used a simulated annealing algorithm to perform a series of perturbations on a randomly chosen LHD until a best possible design is obtained according to the entropy criterion. Many designs had symmetry structure. Ye, William, and Sudjianto (2000) proposed searching for the best design among symmetric Latin hypercube designs. However, instead of randomly searching for a SLHD, they suggested using a columnwise-pairwise exchange algorithm which reduces search time for optimal SLHD. The algorithm involves finding two exchange pairs in a column of the design matrix that result in a design that maintains symmetry as well as has better optimality criterion. In any iteration, there must be two such pairs in order to maintain symmetry of the design. SLHDs exhibit the symmetry property in that any point in the design is reflected through the center of the design space of another point in the design. Such a design is also an OLHD. In addition to the orthogonal property, these designs can have any number of design points independent of the number of variables. The initial SLHD required for the columnwise-pairwise exchange algorithm can be generated from the definition of SLHD. Ye et al. (2000) further found that their algorithm produced consistently better designs than the algorithm of Park (1994). However, the simulated annealing algorithm of Morris and Mitchell (1995) produced even better designs. Even though the CP algorithm does not

produce a true globally optimal design, it is computationally more efficient. Hence, it was used to generate the experimental design for this study.

3.3.1 Optimality Criteria of Experimental Designs

Many optimality criteria for experimental design were presented in the literature. Shanon's entropy criteria was utilized by Shewry and Wynn (1987) to obtain designs with maximum entropy. Entropy is proportional to the amount of information contained in the system. The motivation for the entropy criteria comes from the Gaussian process model. The model is trained in a way so as to reduce uncertainty in the response obtained from the model, which is equivalent to minimizing the posterior entropy.

The minimax and maximin distance criteria were introduced by Johnson, Moore, & Ylvisaker (1990). The minimax criterion tries to minimize the maximum distance between any two points in the design. On the other hand, the maximin distance criterion selects a design that maximizes minimum distance between any two points in the design. These criteria ensure that no point in the design is too far from any other point and achieves uniformity in the distribution of points in the design space. The authors also showed that choosing a design based on the minimax or maximin criterion is same as choosing a design that maximizes the entropy criteria.

The good design should be such that its points are as uniformly distributed as possible in the design domain. The designs with the aim of achieving this uniformity are called uniform designs (Fang, Lin, Winker, & Zhang, 2000). Discrepancy of a design is a measure of uniformity, which is the amount of deviation from uniform distribution of the design points. An optimal design should cover the design space as uniformly as possible so as to achieve good prediction accuracy for a trained model (Fang, Lin, Winker, &

Zhang, 2000). Hence, a lower discrepancy design has a better space-filling property. If $\mathcal{d} = [\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_n]$ is a design in the s-dimensional unit cube design space, $D = C^s$, $[0, \vec{x})$ is a rectangle in this design space, and $N(\mathcal{d}, [0, \vec{x}))$ is the number of points in the rectangle $[0, \vec{x})$, then the following expression is the discrepancy from the uniform distribution at point $\vec{x}$.

$$\left|\frac{N(\mathcal{d},[0,\vec{x}))}{n} - Vol([0,\vec{x}))\right| \tag{3.22}$$

The average of the L_p norm of the discrepancy on C^s is the L_p-discrepancy given by expression in equation 3.23 (Hickernell, 1998).

$$D_p = \left[\int_{C^s} \left|\frac{N(\mathcal{d},[0,\vec{x}))}{n} - Vol([0,\vec{x}))\right|^p\right]^{1/p} \tag{3.23}$$

The authors stated that the L_p-discrepancy does not guarantee a uniform distribution in lower-dimensional space and they suggested three measures of uniformity – symmetric L_2-discrepancy, centered L_2-discrepancy, and modified L_2-discrepancy. They gave a closed form analytical expressions to evaluate each of these. The modified L_2-discrepancy is an improvement over the L_2-discrepancy in that it measures uniformity on lower-dimensional spaces as well. The symmetric L_2-discrepancy measures uniformity of design if any coordinate of a design point is reflected through the mid-point of its range. The centered L_2-discrepancy is invariant under reflection around a plane passing through mid-point if the design space. The expression for L_2-discrepancy is given by equation 3.24 (Hickernell, 1998).

$$CD^2 = \left(\frac{13}{12}\right)^s - \frac{2}{n}\sum_{k=1}^{n}\prod_{j=1}^{s}\left[1 + \frac{1}{2}\left|x_{kj} - 0.5\right| - \frac{1}{2}\left|x_{kj} - 0.5\right|^2\right] +$$

$$\frac{1}{n^2}\sum_{k=1}^{n}\sum_{j=1}^{n}\prod_{i=1}^{s}\left[1 + \frac{1}{2}|x_{ki} - 0.5| + \frac{1}{2}\left|x_{ji} - 0.5\right| - \frac{1}{2}\left|x_{ki} - x_{ji}\right|\right] \tag{3.24}$$

One more measure of uniformity, called the wrap-around L_2discrepancy, is proposed by Hickernell (1998). It measures uniformity of a design when the rectangle in equation 3.22 is no longer anchored around a specific point. Instead, an arbitrary region $[\vec{x}_1, \vec{x}_2)$ is chosen in the design space with wrap-around if either of the points goes outside the design space. An analytical expression is given by Hickernell (1998) in equation 3.25.

$$WD^2 = \left(\frac{4}{3}\right)^s + \frac{1}{n^2}\sum_{k,j=1}^{n}\prod_{i=1}^{s}\left[\frac{3}{2} - \left|x_{ki} - x_{ji}\right|\left(1 - \left|x_{ki} - x_{ji}\right|\right)\right] \qquad (3.25)$$

The discrepancy criteria was used to in this study for optimizing experimental design since a uniform distribution of design points seemed more appropriate and also because of the discrepancy was easy to implement.

3.4 Prediction Accuracy and Regularization

The mean square error of a trained metamodel with respect to training data is always a minimum since the model is developed by minimizing the cost function, which is defined as the sum of squares of the residuals with respect to the training data. However, this measure of the mean square error is not of much value in determining the usefulness of the model for predicting a response at an untried site. In order to determine prediction accuracy, usually, another set of points and their corresponding responses are required. This data set is called the test data. Let $\{y_1, \dots y_m\}$ be the observed response values with mean $\bar{y}_M$ for test data points $\{\vec{x}_1, \dots, \vec{x}_m\}$, $\{\hat{y}_1, \dots, \hat{y}_m\}$ be the predicted response values obtained from the trained metamodel at the test points. Then, the following measures of prediction accuracy can be obtained for the metamodel with respect to the test data (Jin, Chen, & Simpson, 2000).

Mean square error

$$MSE = \frac{1}{m}\sum_{i=1}^{m}(y_i - \hat{y}_i)^2 \tag{3.26}$$

Coefficient of determination

$$R^2 = 1 - \frac{MSE}{\sigma^2} \tag{3.27}$$

Relative average absolute error (RAAE)

$$RAAE = \frac{1}{m}\frac{\sum_{i=1}^{m}|y_i - \hat{y}_i|}{\sigma} \tag{3.28}$$

Relative maximum absolute error (RMAE)

$$RMAE = \frac{\max(|y_i - \hat{y}_i|; i=1,\ldots,m)}{\sigma} \tag{3.29}$$

Here, σ^2 is variance of the observed response, $\sigma^2 = \frac{1}{m}\sum_{i=1}^{m}(y_i - \bar{y})^2$.

These measures of prediction accuracy of a model are relative to the chosen test data. Lower values of MSE, $RAAE$, and $RMAE$ imply better model prediction accuracy. A value of R^2closer to 1 indicates that the model fits the test data closely indicating good prediction accuracy. However, for these statistics to be a true measure of the prediction accuracy, there should be a sufficiently large number of test points and they should be true representatives of the design space and the problem being modeled. In some cases (like the one studied in this thesis), the experiments are expensive and/or time consuming to conduct, thereby making it impossible to obtain a sufficiently large test data set. In such a case, the training data itself can be used to train the model as well as to test the model using a procedure called cross validation (Fang, Li, & Sudjianto, 2006). In order to calculate the cross validation score, the data is split into two sets – the training data set and the test data set. The training data set is used to train the model and the test data set is

used to measure its prediction accuracy. Since the amount of data used for training the model is reduced, the model becomes less accurate. However, in such a case, the test data set can be used to estimate the prediction error, and in the final stage, all of the data can be used as training data to obtain the final model. In this scheme, the prediction error is subjected to which portion of the data is held out for testing. To avoid this bias, *K*-fold prediction measures can be calculated (Fang, Li, & Sudjianto, 2006). In this scheme, the data is partitioned in *K* groups of equal size $\{D_i; i = 1, \dots, K\}$. Data set D_i is held out for determining the prediction error and the remaining data sets are used to train the model. This involves training and testing the model *K* times. The average of the *K* prediction measures is taken as a measure of the model's prediction accuracy. The final model is built using all of the data. The mean square error calculated using the *K*-folds is called *K*-fold cross validation (equation 3.30).

$$CV_K = \frac{1}{K}\sum_{i=1}^{K}\left(\frac{1}{N_i}\sum_{j\in D_i}\left(y_j - \left(\hat{y}_{-D_i}\right)_j\right)^2\right) \tag{3.30}$$

Here, $\left(y_{-D_i}\right)_j$ is the predicted response at point $\vec{x}_j \in D_i$, obtained from a metamodel trained by excluding the D_i part of the data set, y_j is the observed response at $\vec{x}_j$ and N_i is the number of points in the part D_i of the data set. If each fold of the data set contains only one point, then the cross validation score is called leave-one-out cross-validation.

$$CV = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_{-i})^2 \tag{3.31}$$

Here, $\hat{y}_i$ is the predicted response at ith data point given by the metamodel trained by excluding the ith data point.

As discussed in section 3.2.1, the polynomial models trained using ridge regression have a parameter λ to penalize the estimators that achieve a tradeoff between

expectation and variance. In LASSO, it controls the threshold for including or excluding a basis function in the model. For the radial basis functions discussed in section 3.2.2, the attenuation factor, r, controls the spread of each basis function's region of influence. The choice of these constants, called regularization parameters λ, affects the prediction accuracy of the developed metamodel. An optimum λ can be chosen using a procedure called regularization such that it maximizes the prediction accuracy measured by any of the metrics described earlier. Since the metamodel depends on the regularization parameter λ, the metric of prediction accuracy, such as the cross validation score, is also a function of λ. If cross-validation is chosen as the prediction error metric, then this minimization problem can be stated as follows.

$$\hat{\lambda} = \min_{\lambda} CV(\lambda) \tag{3.32}$$

This minimization problem can be solved using an optimization algorithms such as simulated annealing. However, Fang et al. (2006) suggested a heuristic based approach. In this approach, a range is selected for λ and a uniform grid is created in this range. The cross-validation score is evaluated for the chosen model by varying λ from the smallest value to the largest value. If the optimal λ lies in the chosen range, then as λ varies from the smallest to the largest value, the cross-validation score decreases, reaches a minimum at a particular λ, and then increases again. If the minimum is not found in the chosen range, then a different range of λ is chosen and the procedure is repeated. The value of λ at which cross-validation score reaches minimum, is selected as the optimal regularization parameter. Final metamodel should be developed using all of the data and the optimal $\hat{\lambda}$.

In this study, the leave one out cross validation score was used as a heuristic for the heuristic-based regularization parameter selection of Fang et al. (2006) to train a metamodel.

CHAPTER IV

OPTIMIZATION OF THE FFA-W3-301 AIRFOIL WITH CAVITY USING METAMODELS

This chapter explains various aspects of the metamodel development process discussed in chapter III for the problem of optimization of a cavity shape for FFA-W3-301 airfoil.

4.1 Parameterization for the Shape of Cavity on FFA-W3-301 Airfoil

I conducted a preliminary investigation of flow over the NACA0012 airfoil with a backward facing step on the upper surface (like the Kline-Fogleman airfoils) and a positive angle of attack indicated. Results indicated that the vortices forming in the step were swept away by the flow. This resulted in continuous vortex shedding, which was detrimental to the airfoil performance. So, in order to promote the formation of a stable vortex, an upwardly concave NURBS curve was used to define the shape of the cavity and a separate NURBS curve was used to define the aft portion of the airfoil. The cavity curve requires a start point, an end point, a tangent vector at each of these locations, and a weight for each tangent that determines the extent to which the curve is stretched in the direction of the tangent. The rear portion of the airfoil including the cavity and the cavity design parameters is shown in figure 4.1. The curves C_1 and C_2 redefine the upper surface of the airfoil. Though the start point of C_1 lies on the airfoil surface, its end point does not if a non-zero offset x_7 is used. Because of this, a separate NURBS curve was used for the

aft portion of the airfoil. The parameters and their description are given in table 4.1. Only parameters x_1 to x_7 were varied for this study while parameters x_8 to x_{11} were held constant. The parameters x_1 and x_6 determine the start and end positions of the cavity, respectively. Changing these parameters changes the location and length of the cavity. Varying these parameters will give insight into the effect of position and length of the cavity on the airfoil performance. The parameters x_3 and x_5 determine the front and back depth of the cavity and parameters x_4 and x_2 affect the direction of fluid flow into and out of the cavity at its ends. Together, parameters x_1 to x_6 determine the cavity's position, length, and depth. The parameter x_7 is the amount by which the end of the cavity is lifted above the original airfoil surface. A larger x_7 causes the cavity to protrude more into the flow thereby trapping more energy in to the vortex inside the cavity at the same time possibly causing the flow separate earlier on the aft position of the airfoil. The parameters x_1 to x_7 form a seven-dimensional design space. Initial simulations using randomly chosen cavity parameters indicated that the cavity should be in the aft portion of the airfoil upper surface. Hence the range of each parameter value was chosen as shown in table 4.2. The values of parameters $x_1, x_3, x_5, x_6,$ and x_7 are in fractions of the chord length while the angles x_2 and x_4 are in degrees.

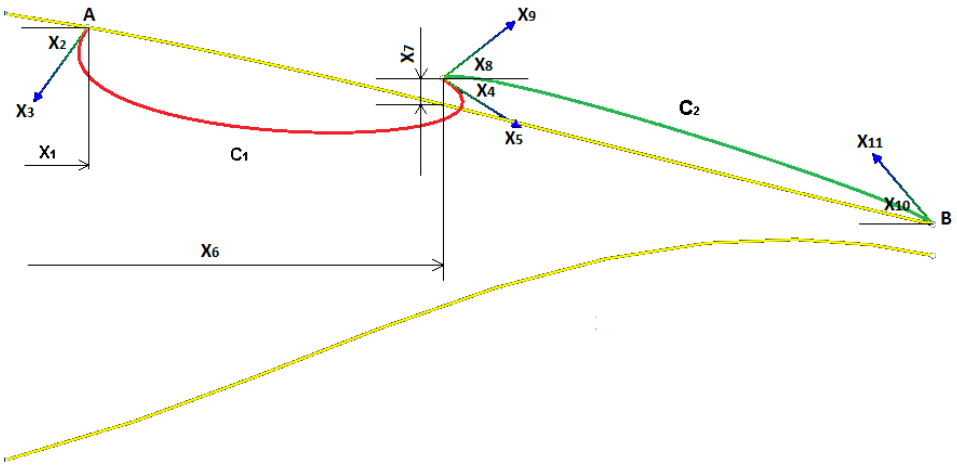


Figure 4.1 The cavity design parameters

Table 4.1 Description of parameters for the cavity design

Parameter	Description
x_1	Distance of start point of curve C_1 from leading edge of the airfoil along its chord
x_2	Angle made by start vector of curve C_1 with negative x axis in degrees
x_3	Scale factor for start vector of curve C_1
x_4	Angle made by end vector of curve C_1 with positive x axis in degrees
x_5	Scale factor for end vector of curve C_1
x_6	Distance of end point of curve C_1 from leading edge of the airfoil along its chord
x_7	Vertical distance of end point of curve C_1 from the point on original airfoil curve at distance x_6 from leading edge of the airfoil
x_8	Angle made by start vector of curve C_2 with positive x axis
x_9	Scale factor for start vector of curve C_2
x_{10}	Angle made by end vector of curve C_2 with negative x axis
x_{11}	Scale factor for end vector of curve C_2

Table 4.2 Range of experimental design domain for FFA-W3-301 cavity parameters

Parameter	Start	End
x_1	0.5	0.6
x_2	22.5	67.5
x_3	0.05	0.1
x_4	22.5	67.5
x_5	0.05	0.1
x_6	0.65	0.75
x_7	0.0	0.02

4.1.2 Software for Design of Airfoil Profile with Cavity

To generate the profile of an airfoil with a cavity based on the parameterization described above, a MATLAB code with a graphical user interface was developed that takes as inputs a xy file that defines the original airfoil profile, the cavity parameter values and other parameters for controlling the NURBS curve generation. Figure 4.2 shows the user interface of the application. Clicking on the "Read" button opens a dialog box that allows selection of the xy data file for the airfoil profile. This file must have two columns, the first column for the x coordinate and the second column for the y coordinate of the points that define original airfoil shape. The coordinates in this file must start from the trailing edge on lower surface of the airfoil section, wrap around the airfoil profile from the leading edge and end at the trailing edge on upper surface of the airfoil.

First, the software calculates the original airfoil shape by generating a C^2-continuous NURBS curve that passes through the points read from the airfoil definition file. The NURBS curve generation is controlled using settings specified in the "Parameterization", "Boundary Conditions", "np" (number of points), "tol" (tolerance), and "iter" (iterations) boxes. Once the NURBS curve is generated, the NURBS curve is

trimmed at the point on upper surface of the airfoil at distance “Start Pos” from leading edge of the airfoil until the trailing point on the upper airfoil surface. Then, two C^2-continuous NURBS curves C_1 and C_2 are generated using the cavity parameters as mentioned above. The values of these cavity parameters should be specified in the text boxes next to the parameter names in the “Cubic Curve” box on the GUI. The resulting airfoil profile with the cavity is then displayed on the user interface. The control points for the cubic curve of the cavity can be made visible by selecting the toggle button “Show control points” at the bottom of the user interface. The resulting airfoil shape with the three NURBS curve can be saved in the .NC file format (Jiang & Remotigue, July 1998) by clicking the “Write” button and specifying file name in the save dialog box. The .NC file can be directly read into GUM-B for grid generation purpose.

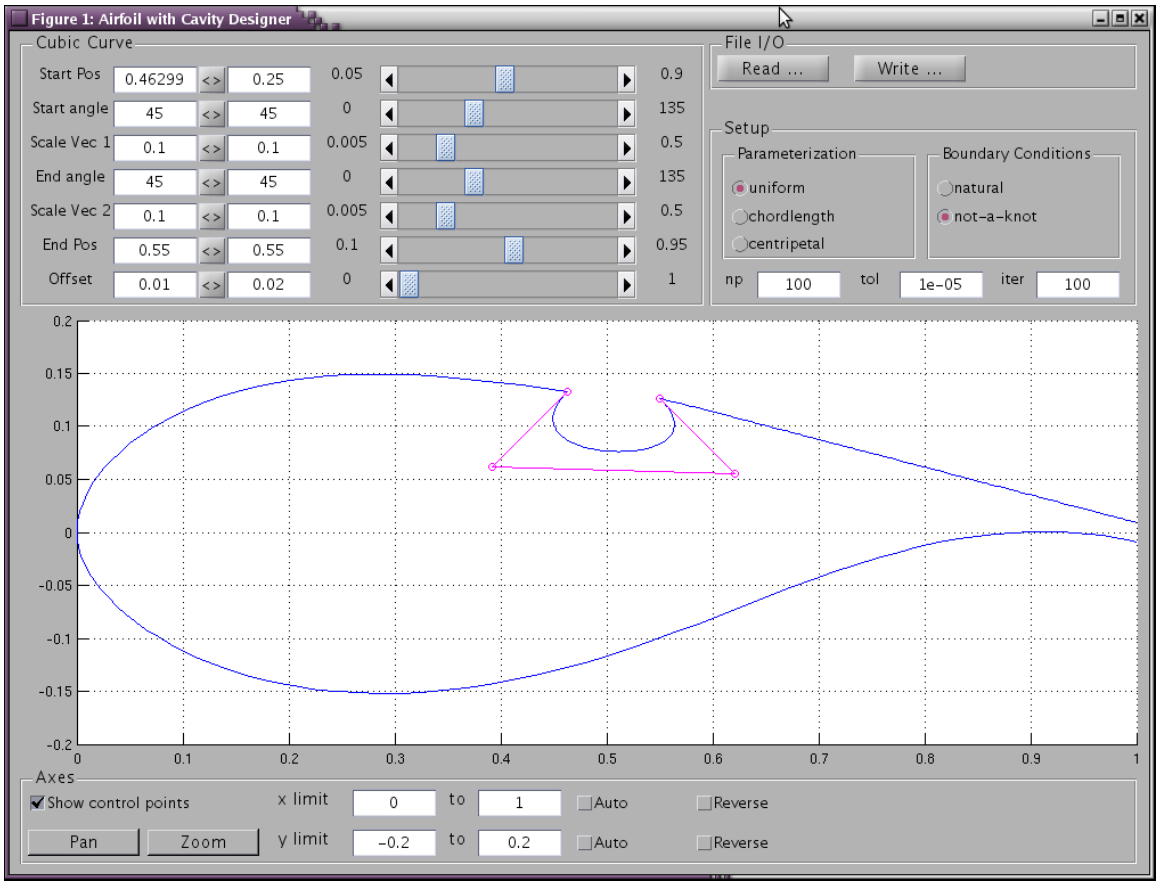


Figure 4.2 The MATLAB user interface of the application for generating airfoil profile with cavity

Since the grid is a structured grid, the grid in the cavity requires a domain decomposition as shown in figure 4.3. This avoids cells with large cell to face angles making the CFD simulations more accurate. The grid shown in the figure is coarse for visualization purpose. However, for actual simulations, the grid in the cavity was much more refined. The grid surrounding the airfoil is modified accordingly.

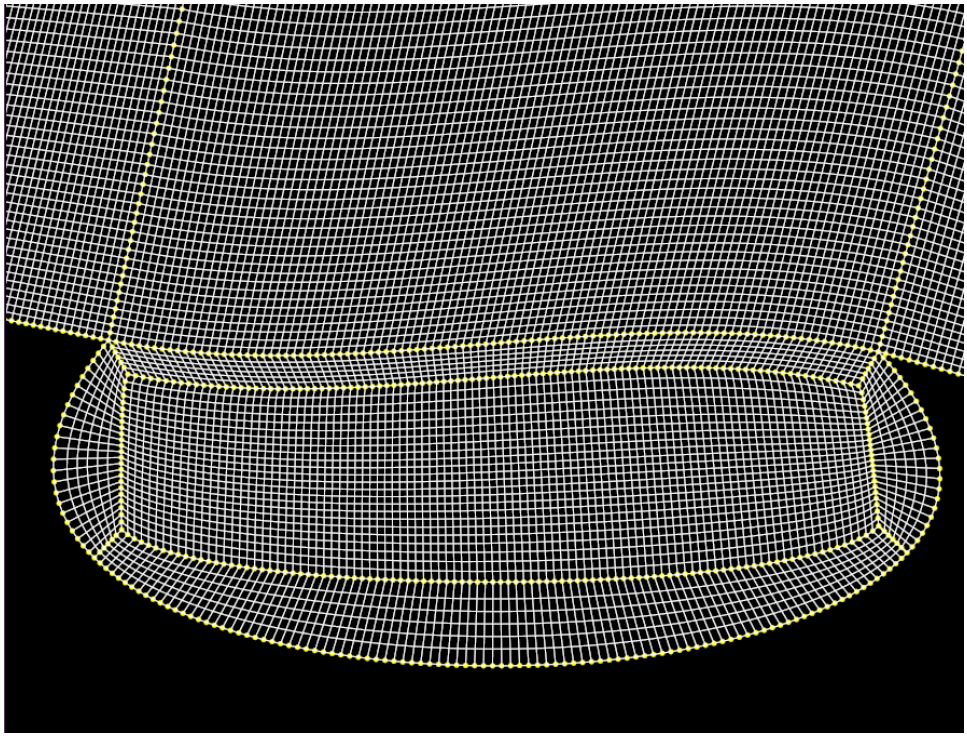


Figure 4.3 Structured grid in the cavity

4.2 Design of Experiment for the Cavity Parameterization

For generating the experimental design, a MATLAB program was developed which can generate OLHD or SLHS with centered L_2-discrepancy as the optimality criteria. The graphical user interface of the software is as shown in figure 4.4. To generate a design, click on the "Experiment" menu and then "New" menu item. This opens a dialog box as shown in figure 4.4. Select the required type of design - "SLHD" or "OCLHD". Then specify the number of iterations. These are the number of Latin hypercube designs searched for best possible design. For OLHD, the number of design points are fixed depending on the number of design variables. On the other hand, for SLHD, the number of design points should be specified in the field "n". The factors and

their ranges are specified in the dialog box. On clicking the “Ok” button, the software starts the process of generation of optimal design and chooses the best possible design among all the explored designs. The software internally uses the columnwise-pairwise exchange algorithm (Ye, William, & Sudjianto, 2000) for optimization of LHDs. The optimal design, along with its centered discrepancy, is displayed in the main window. The software also allows saving the generated design and opening an existing design generated by this software. If a design is already opened, it can be extended by keeping the original design points and adding more design points such that the type of design (SLHS or OLHD) remains same. This can be achieved by using the “Extend” menu item under the “Experiment” menu. This feature is useful in a situation when training data has already been collected for a design but was found to be too scarce to produce sufficiently accurate information to train a metamodel. Generating a new design with more points will waste the effort already spent in gathering training data on the earlier design unless the new design adds points while maintaining the training points in the original experimental design. This extension should be done in a way such that the structural property of the original design remains intact. The centered discrepancy code was verified using LHDs with the known value of centered discrepancy mentioned in (Fang, Li, & Sudjianto, 2006). For modeling the performance of the FFA-W3-301 airfoil cavity, a 67 point symmetric Latin hypercube design was constructed in the design space of the seven design variables listed in table 4.2. The design points are listed in table B.1.

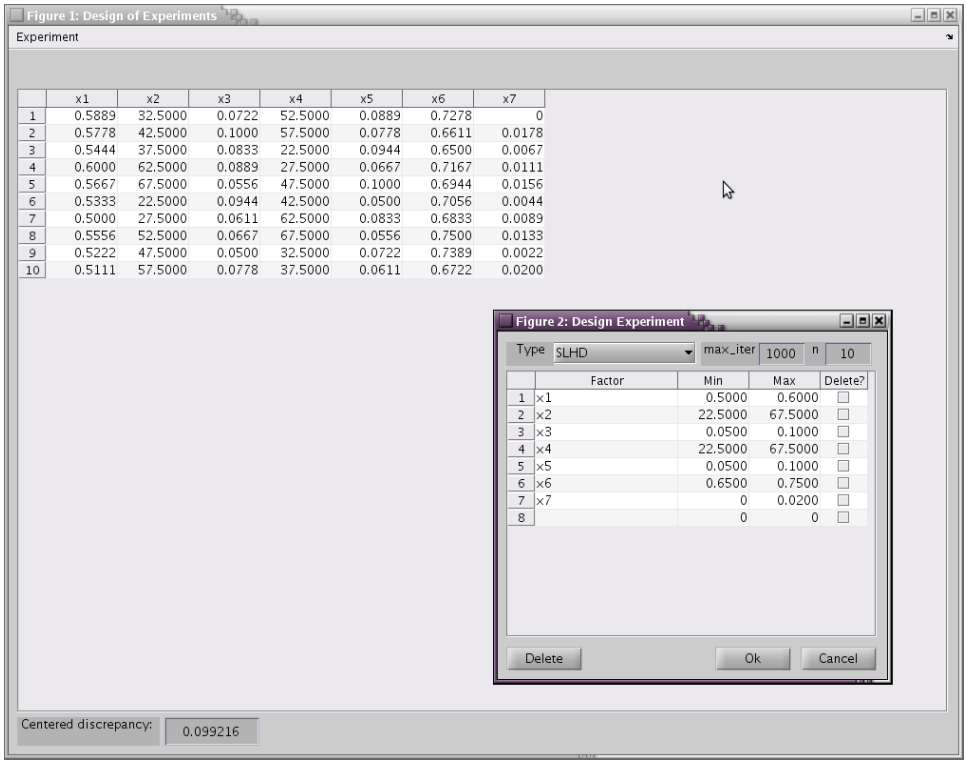


Figure 4.4 Graphical user interface of application for generating SLHD and OLHD

4.3 Software for Building Metamodels

A C++ code was developed for generating a regularized metamodel from a given experiment and training data set. Input to the program is an XML file containing training data and specifications of metamodels to train. The program trains specified metamodels and writes the trained model parameters into a file. The code has an API that allows reading a saved metamodel. Once it is read and loaded into memory, the response as well as the gradient of the metamodel at any point in the design space can be obtained. So, a model is generated once and reused later in the optimization routines. The gradient of the modeled response at a point is useful in driving optimization routines, which is necessary to find the optimal cavity shape as explained in later section.

It is important to validate the program by using it to model some toy functions. For this purpose, two functions were considered. The details of the numerical experiments are given in appendix A.

4.4 Development of Metamodel of the Performance of the FFA-W3-301 Airfoil with Cavity

In this study, performance of the FFA-W3-301 airfoil with a cavity was measured by its lift coefficient (C_l), drag coefficient (C_d) and the ratio of the two coefficients (C_l/C_d) at 12°. The 12° angle of attack was chosen because, at this angle (and higher angles), the original airfoil showed a large separated region that resulted in stall and the objective of this study was to determine a passive cavity configuration that alleviates this problem. The experimental design of 67 points was used to generate 67 airfoil profiles of the FFA-W3-301 airfoil with a cavity. Each of these airfoil shapes was used to generate a structured grid as discussed earlier and Loci/CHEM simulations were performed to obtain C_l and C_d as response values for the computer experiments. Since the amount of effort in collecting the lift and drag coefficients data for each cavity configuration was large, a small test data set of 10 points was generated. Though this test data set was not sufficient enough to give an accurate estimate of the prediction accuracy of the metamodels, it was used to get a rough estimation of the prediction accuracy. The performance metrics of the metamodels (as discussed in section 3.4 of chapter III) with reference to this test data are listed in table 4.3 and 4.4.

Table 4.3 Test statistics for the radial basis function metamodels developed for the lift coefficient of the FFA-W3-301 airfoil with cavity

Type of RBF Metamodel	CV Score $\times 10^{-4}$	MSE $\times 10^{-4}$	R^2	RAAE	RMAE
Gaussian	3.470847	5.24253	0.916088	0.243886	0.457296
Multiquadratic	3.491105	4.40783	0.929448	0.235904	0.430594
Inverse Multiquadratic	3.406817	4.57445	0.926781	0.236947	0.423978
Inverse Quadratic	3.394605	4.64829	0.925600	0.237544	0.422998

Table 4.4 Test statistics for the radial basis function metamodels developed for the drag coefficient of the FFA-W3-301 airfoil with cavity

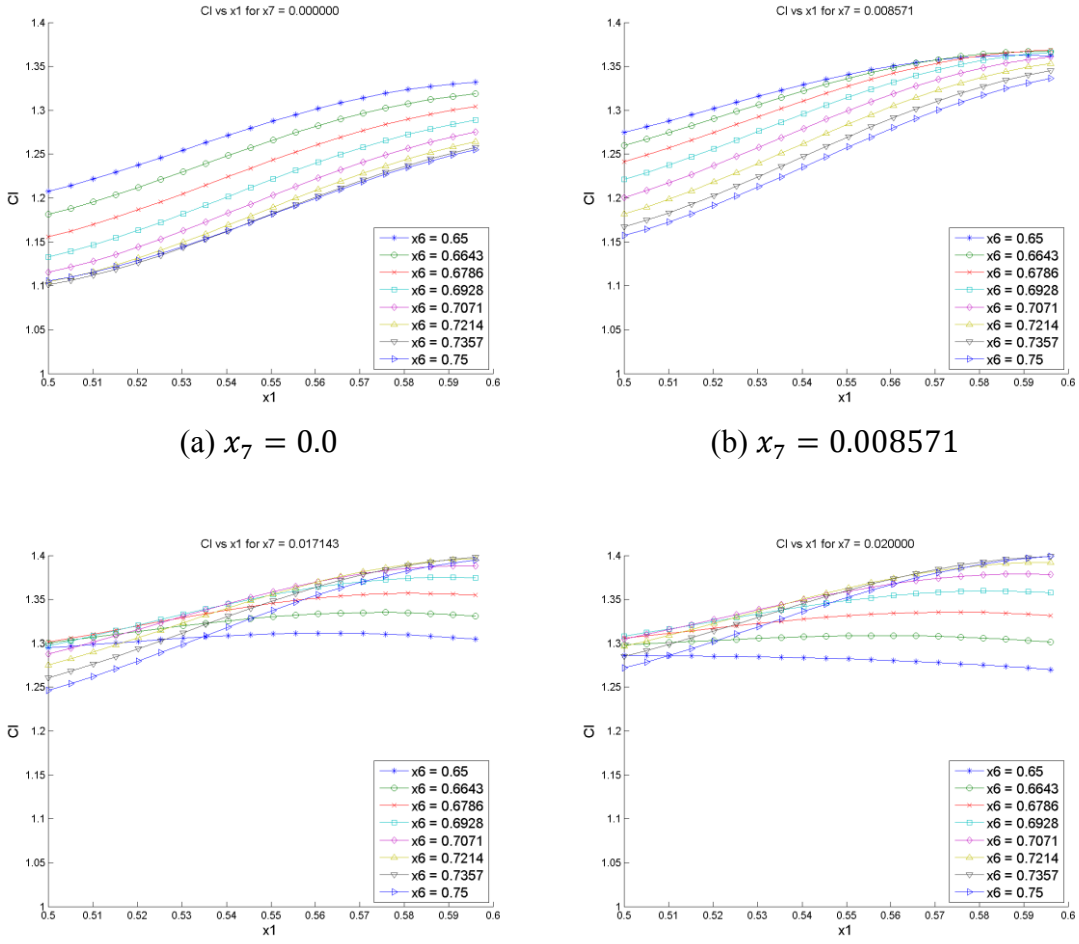
Type of RBF Metamodel	CV Score $\times 10^{-7}$	MSE $\times 10^{-6}$	R^2	RAAE	RMAE
Gaussian	6.173603	2.38449	0.925052	0.228254	0.431922
Multiquadratic	7.295857	2.44354	0.923197	0.232886	0.437017
Inverse Multiquadratic	6.896605	2.36219	0.925754	0.22927	0.420819
Inverse Quadratic	6.775923	2.35108	0.926103	0.228686	0.416803

The R^2, RAAE and RMAE values for the lift and drag coefficient data indicate that the drag and lift coefficients predicted by the models can have errors as large as ~45% of the standard deviation in observed response values. Though this is a large error for the lift and drag coefficients of an airfoil, it should be kept in mind that the use of these metamodels is to enable design space exploration for finding an optimum configuration. As long as a metamodel captures the general shape of the response surface, the errors in the predicted response given by the metamodel should not interfere with the conclusions of this work.

The purpose of this study is to examine the design space of the cavity shape parameters to find a cavity configuration that gives maximum lift and minimal drag. Therefore the objective function formulation should include both the lift and drag coefficient. However, only the lift coefficient is considered for the optimization reported here.

4.5 Response Surface of Lift Coefficient

The design space of the cavity shape parameters is seven dimensional. It is hard to visualize a response surface that is more than 2 dimensional. Therefore attention was focused on the dominant parameters. A linear polynomial model was developed for the lift coefficient using ordinary least square regression. The model had $R^2 = 0.744424$. Even though the model was a very rough approximation of the true response, its standardized regression coefficients gave insight into the parameters that had a significant effect on the response. It was found from this model that the airfoil performance is most sensitive to x_1, x_6, and x_7, that is, the start and end positions of the cavity, and amount by which the end of the cavity protrudes into the flow. The multiquadratic radial basis function metamodel was used to plot lift coefficient against x_1 for various values of x_6 and x_7 keeping other design parameter values at the midpoint of their domain. The plots are shown in figure 4.5. These plots indicate that for low values of x_7, the lift coefficient increases if x_1 increases and x_6 decreases. That is, if the cavity length becomes smaller, the lift coefficient increases. However, for higher values of x_7, increasing both x_1 and x_6 increases lift coefficient. This shows that there is a possibility of improving lift by moving the cavity rearwards on the airfoil upper surface and increasing x_7.



(a) $x_7 = 0.0$ (b) $x_7 = 0.008571$

(c) $x_7 = 0.017143$ (d) $x_7 = 0.2$

Figure 4.5 Plot of response surface of the multiquadratic metamodel of lift coefficient of the FFA-W3-301 airfoil with cavity at 12° AoA.

Each plot is lift coefficient vs x_1 graph for a fixed value of x_7.

4.6 Optimization of the Cavity Shape and Incremental Model Improvement for 12° Angle of Attack

The drag and lift coefficient radial basis function metamodels for 12°AoA have R^2 more than 90% indicating that the models have moderate accuracy. Improving the global accuracy of the models requires adding more training points in the design domain.

Since the process of collecting response values is time consuming, an incremental approach was used in this study. For the initial model, its optimum point was found using an optimization algorithm. The airfoil with the cavity was designed for this optimum point and then simulated using Loci/CHEM to compute the lift and drag data for the configuration. It was compared with the values given by the model. If the error was large, that optimal point was used to augment the design matrix and to build a new improved model. These steps were repeated until the lift and drag coefficient values given by the model and the simulation matched.

In this study, the multiquadratic radial basis function model was used to perform the incremental optimization cycles. For optimization, the Matlab optimization toolbox with the trust-region reflective algorithm was used. The optimized cavity that gave highest lift to drag ratio for FFA-W3-301 airfoil at 12° angle of attack is defined by parameter values listed in table 4.5. This cavity was designated as o19. The FFA-W3-301 airfoil with the o19 cavity is shown in figure 4.6.

Table 4.5 The o19 cavity parameter values

Parameter	**Value**
x_1	0.657382
x_2	55.955368
x_3	0.089359
x_4	46.588125
x_5	0.095150
x_6	0.764157
x_7	0.016272

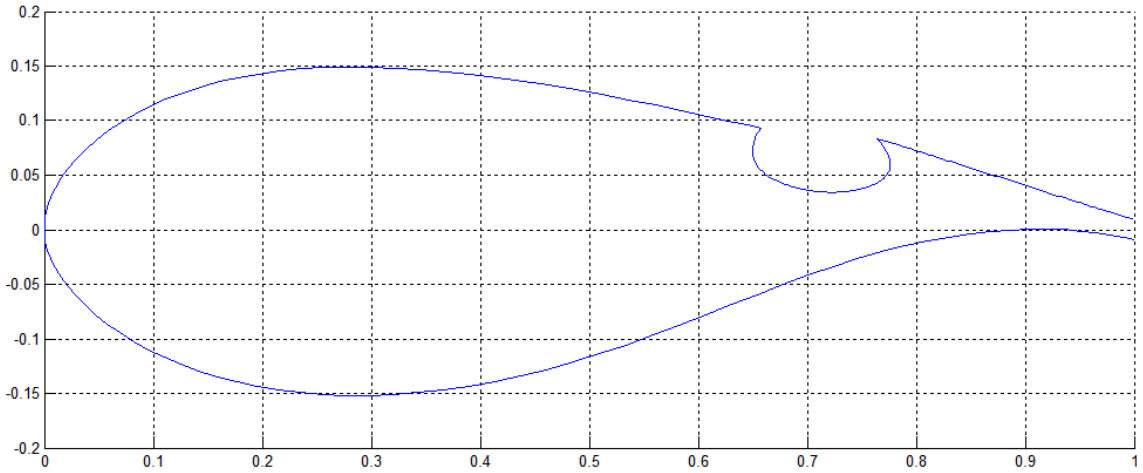
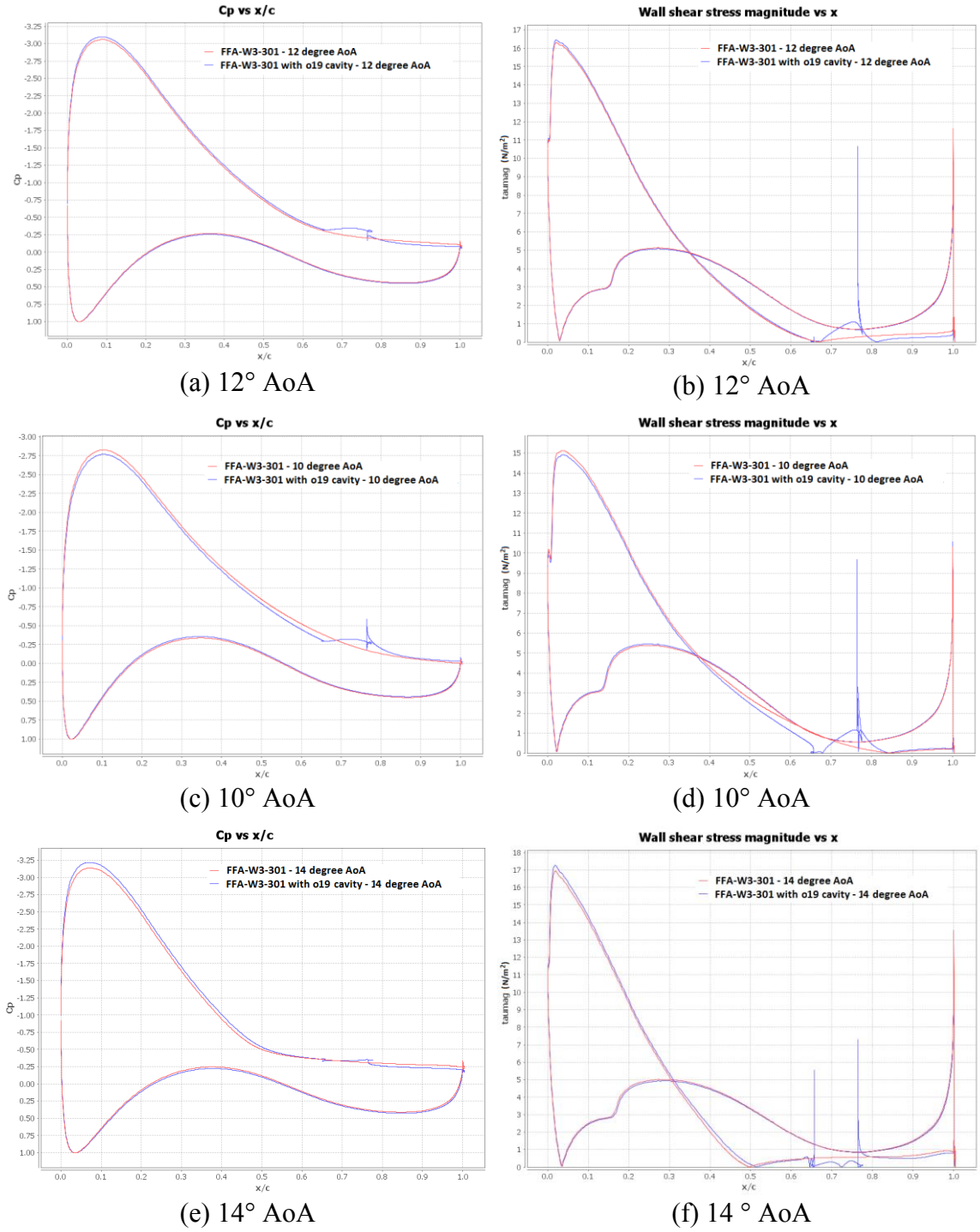


Figure 4.6 The FFA-W3-301 airfoil with the o19 cavity

Table 4.6 Lift and drag coefficients for FFA-W3-301 and FFA-W3-301 with the o19 cavity for 10°, 12°, and 14° AoA

AoA	**Performance metric**	**FFA-W3-301**	**FFA-W3-301 with o19 cavity**
10°	C_l	1.294889	1.256315
	C_d	0.0378466	0.0395740
	C_l/C_d	34.2141	31.7460
12°	C_l	1.376509	1.410069
	C_d	0.0487612	0.0473990
	C_l/C_d	28.2296	29.7489
14°	C_l	1.353612	1.404948
	C_d	0.0723395	0.0645478
	C_l/C_d	18.7120	21.7660

The FFA-W3-301 airfoil and the modified FFA-W3-301 airfoil with the o19 cavity has lift and drag coefficients as listed in table 4.6. The pressure coefficient and the wall shear stress magnitude graph comparisons are presented in figure 4.7.

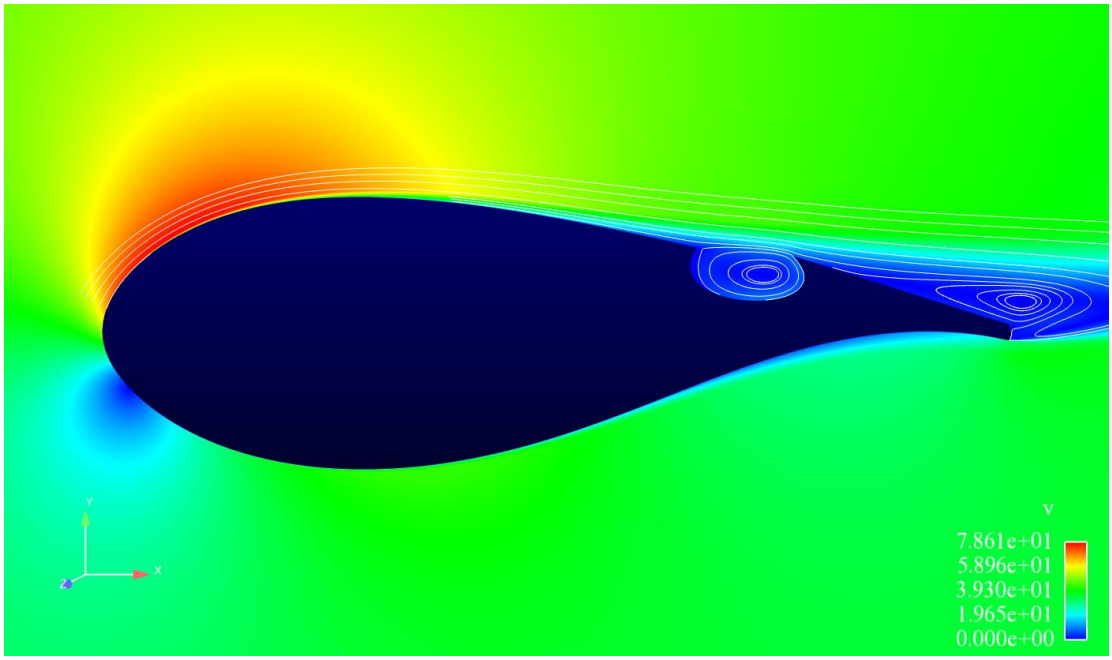


(a) 12° AoA (b) 12° AoA

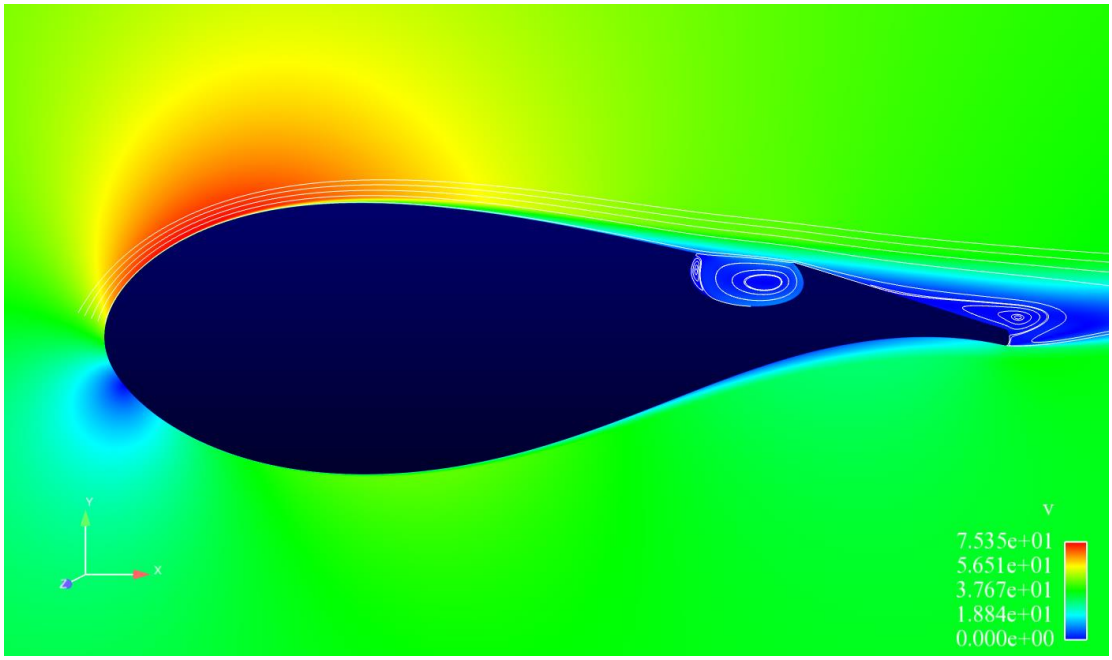
(c) 10° AoA (d) 10° AoA

(e) 14° AoA (f) 14 ° AoA

Figure 4.7 Graphs comparing pressure coefficient (Cp) and wall-shear stress magnitude between FFA-W3-301 and FFA-W3-301 with the o19 cavity at various angles of attack.

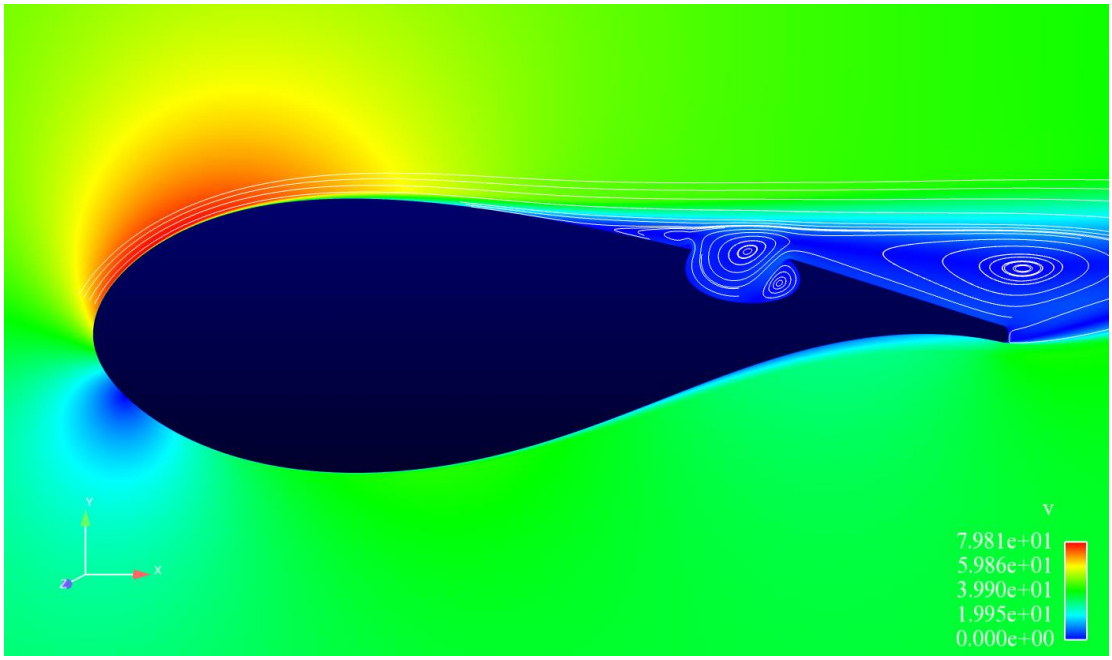


(a) 12° AoA



(b) 10° AoA

Figure 4.8 Streamlines of flow over the FFA-W3-301 airfoil with the o19 cavity at various angles of attack



(c) 14° AoA

Figure 4.8 (continued)

The pressure coefficient graph for a 12° angle of attack shows that for the airfoil with the o19 cavity, there is a small "suction" inside the cavity. Also the wall-shear stress magnitude graph shows that the flow stays attached until 0.81 of chord after the cavity ends at 0.764157 of chord. Because of this partial reattachment, the o19 cavity configuration produces a higher lift to drag ratio.

The same cavity was tested for 10° and 14° angles of attack. For the 10° AoA, shown in figures 4.7c, 4.7d and 4.8b, the cavity started upstream of the point of separation (~0.845) on the unmodified FFA-W3-301 airfoil. This resulted in a higher suction-side pressure and hence a lower lift. However, it can be seen from the wall-shear stress magnitude graph that the flow stays attached for a greater distance downstream of

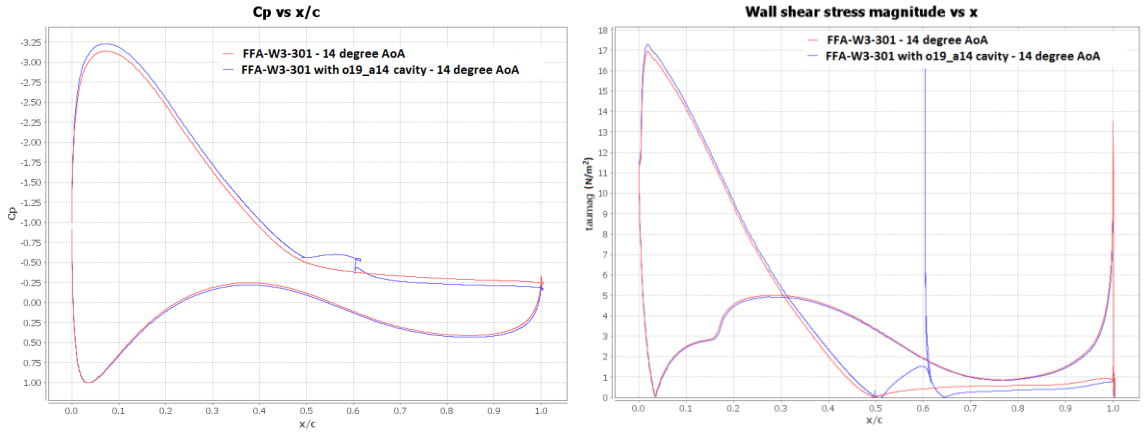
the cavity. For the 14° AoA, shown in figures 4.7e, 4.7f and 4.8c, the cavity starts after the unmodified airfoil's point of separation and is seen to be completely in the separated region. But it results in a higher lift to drag ratio due to a lower suction-side pressure and better pressure recovery after the cavity. The flat pressure coefficient graph inside the cavity and in the aft region shows that the flow is separated. The wall-shear stress magnitude graph shows that there is no reattachment of the flow.

4.7 The o19 Cavity at Point of Separation

In the 12° AoA case, the o19 cavity starts at the point of separation. However, it does not start at the point of separation in the case of 10° and 14 ° AoA. Hence, a new class of cavity was used to test the airfoil performance under different angles of attack. This class of cavity was based on the o19 cavity parameter values, except the x_1 parameter value was set to the x-coordinate of point of separation for each particular angle of attack and x_6 was adjusted such that the cavity length, $(x_6 - x_1)$, stayed the same as the o19 cavity. Such a cavity was named, o19_aXX, where the suffix aXX indicates the angle of attack for which the cavity was "designed". For example, the o19_a14 cavity for the FFA-W3-301 airfoil has the same parameter values as the o19 cavity except x_1 was set to the x-coordinate if the point of separation from the leading edge of the airfoil for 14° AoA and x_6 was adjusted accordingly to keep the same cavity length. The o19_a14 cavity was tested for 14° AoA. The results are as shown in table 4.7 and figure 4.9.The wall-shear stress magnitude graph shows partial reattachment similar to the FFA-W3-301 with the o19 cavity for the 12° AoA case. Also the lift to drag ratio was higher than the fixed position o19 cavity at 14° AoA. Figure 4.10 shows the FFA-W3-301 airfoil with the o19_a14 cavity.

Table 4.7 Lift and drag coefficients for the FFA-W3-301 airfoil and the FFA-W3-301 with the o19_a14 cavity for 14° AoA

AoA	Performance metric	FFA-W3-301	FFA-W3-301 with o19_a14 cavity
14°	C_l	1.353612	1.417695
	C_d	0.0723395	0.0637078
	C_l/C_d	18.7120	22.2531



(a) (b)

Figure 4.9 Graphs comparing pressure coefficient (Cp) and wall-shear stress magnitude of the FFA-W3-301 airfoil with those of the FFA-W3-301 airfoil with the o19_a14 cavity at 14° AoA

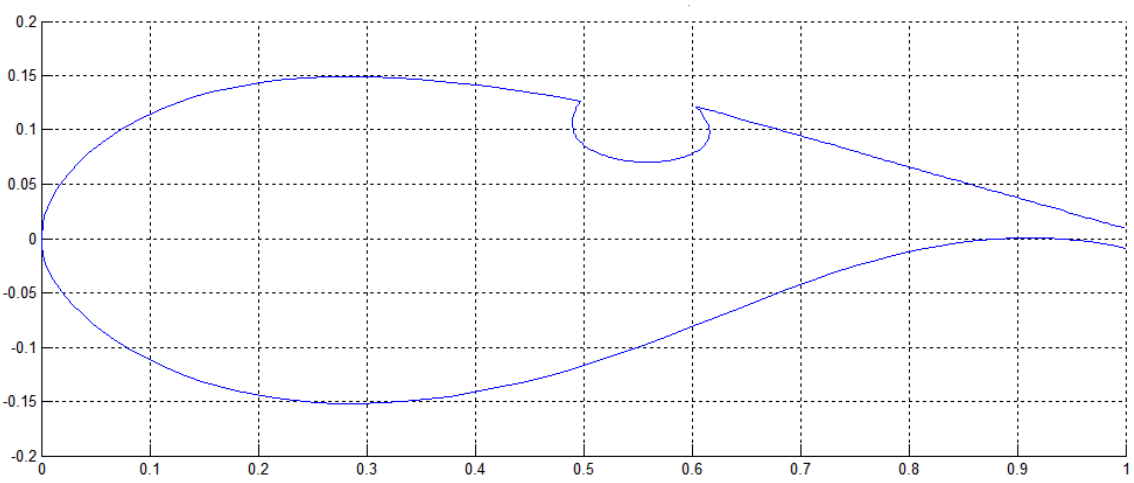


Figure 4.10 The FFA-W3-301 cavity with the o19_a14 cavity

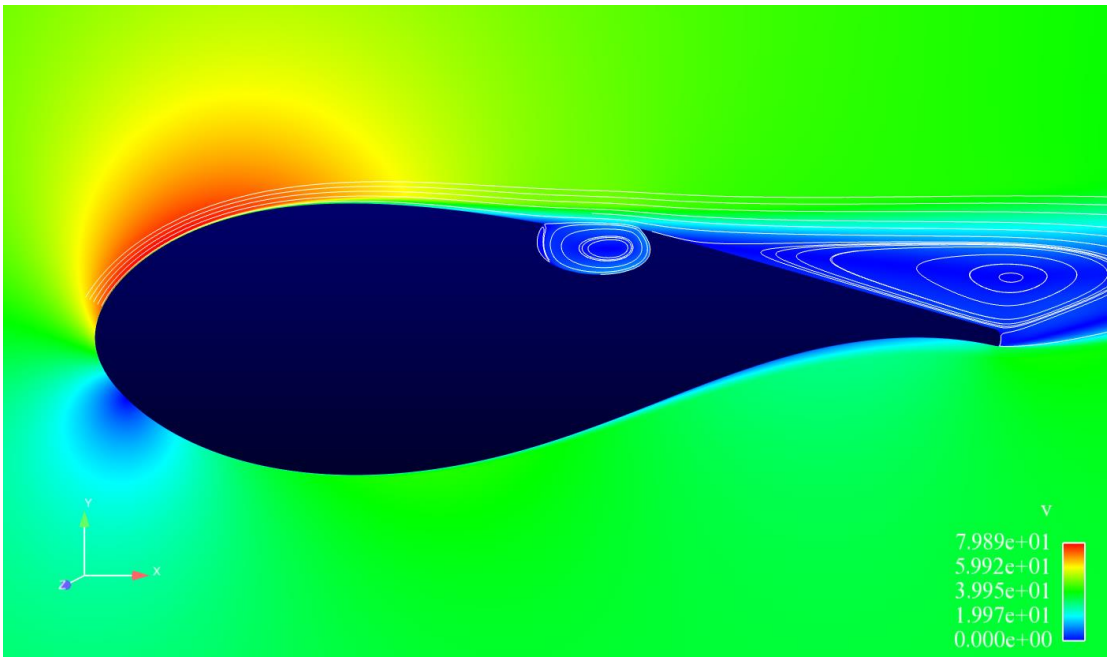


Figure 4.11 Streamlines of flow over the FFA-W3-301 airfoil with the o19_a14 cavity at 14° AoA

4.8 Testing of the o19_aXX Cavity on NACA0024 Airfoil

An NACA0024 airfoil was used to test the o19_aXX cavity at 12° and 14° AoA. Loci/CHEM simulations were performed on the NACA0024 airfoil. At 12° AoA, the point of separation was located at a distance of around 78% of the chord. By placing the o19 cavity at this location, NACA0024 airfoil with the o19_a12 cavity was obtained (figure 4.13(a)). For 14° AoA, the point of separation was at around 68% of the chord. Therefore, the o19_a14 cavity configuration was obtained by placing the o19 cavity at 68% of the chord on the airfoil's upper surface (figure 4.13(b)). These modified airfoils were tested for 12° and 14° AoA, respectively. Improvement in the lift to drag ratio was around 3% at 12° AoA and 6% at 14° AoA. Since the NACA0024 airfoil is a thinner airfoil compared to the FFA-W3-301 airfoil, the separation occurred at a larger chord distance compared to the FFA-W3-301 airfoil if they both are subjected to the same angle of attack. An interesting observation is that, for the case of flow over the FFA-W3-301 airfoil at 12° AoA, flow separation occurred at around 66% of the chord. In case of flow over the NAC0024 airfoil at 14° AoA, flow separation occurred at around 68% of chord. In terms of the length of the separated region, both these cases are similar. Further, the AoA adaptive o19 cavity in these two cases showed similar percentage improvement in L/D. Therefore, it seems that the percentage improvement in the L/D ratio of a stalled airfoil is proportional to the length of the separated region. This is probably the reason that the improvement in case of the NACA0024 airfoil was not as good as the FFA-W3-301 airfoil at 12° AoA. Improvement in the lift and drag coefficient for both the 12° and 14° AoA case is shown in table 4.8 and 4.9. Figure 4.12 shows comparison of the pressure coefficient between the NACA0024 airfoil and the NACA0024 airfoil with the

o19_a14 cavity at 14° AoA. This graph shows similarity to the case of the FFA-W3-301 airfoils with the o19_aXX cavity in that, the trapped-vortex cell produced slight suction inside the cavity. Also the lower pressure was achieved on the suction side of the airfoil, thus producing higher lift.

Table 4.8 Lift and drag coefficients for the NACA0024 and the NACA0024 with the o19_a12 cavity at 12° AoA

AoA	Performance metric	NACA0024	NACA0024 with o19_a12 cavity
12°	C_l	0.960850	0.981623
	C_d	0.0298366	0.0291782
	C_l/C_d	32.2037	33.6423

Table 4.9 Lift and drag coefficients for the NACA0024 and the NACA0024 with the o19_a14 cavity at 14° AoA

AoA	Performance metric	NACA0024	NACA0024 with o19_a14 cavity
14°	C_l	1.067603	1.095226
	C_d	0.0379768	0.0368712
	C_l/C_d	28.1119	29.7041

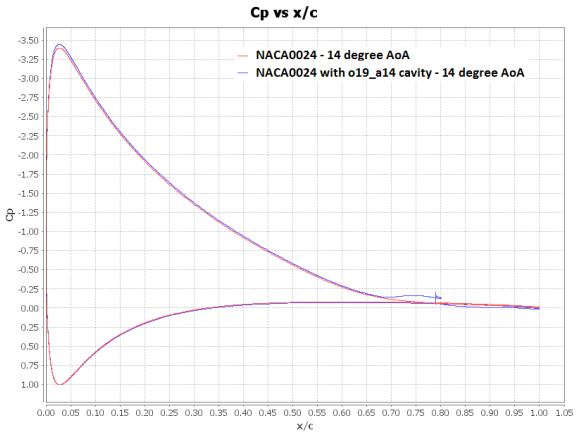
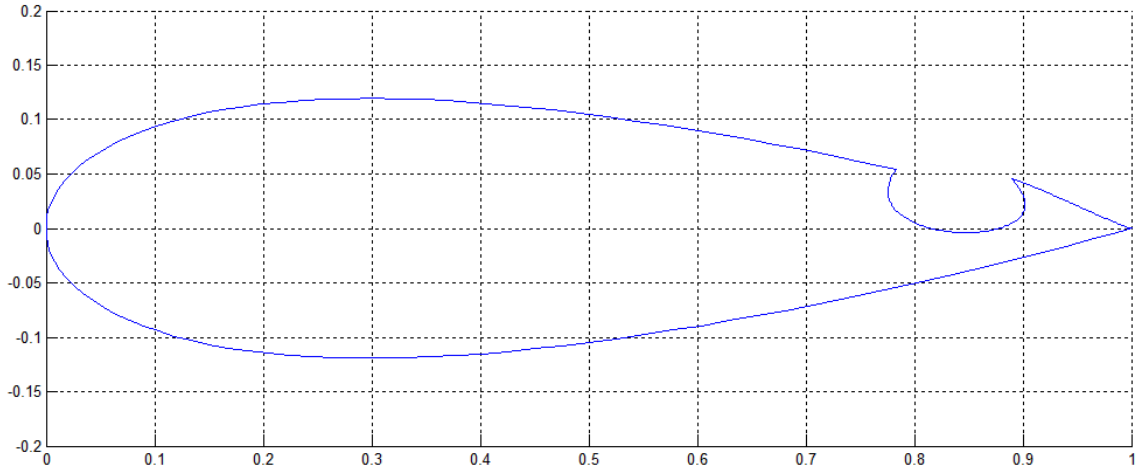
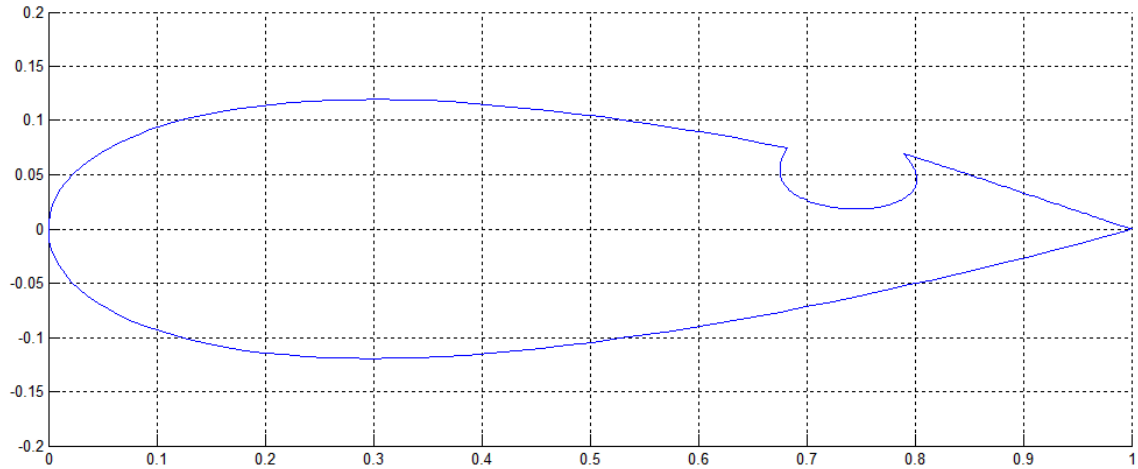


Figure 4.12 Comparison between pressure coefficient of the NACA0024 airfoil and the NACA0024 airfoil with the o19_a14 cavity at 14° AoA



(a) NACA0024 airfoil with the o19_a12 cavity



(b) NACA0024 airfoil with the o19_a14 cavity

Figure 4.13 Profile of the NACA0024 airfoil with the o19_a12 and the o19_a14 cavities

CHAPTER V

CONCLUSION

In conclusion, although no modified airfoil configuration incorporating a passive trapped-vortex cavity was found to increase L/D for all airfoil attitudes, the findings of this effort have clearly shown that cavity shape does appear to affect whether or not a passive cavity has a beneficial effect. Specifically the parameters x_1, x_6, and x_7 were found to have the largest impact. In this work, an optimization of these parmeters has led to a cavity design referred to as cavity o19 which yields a 5% improvement in L/D (notably by increasing lift and decreasing drag) for FFA-W3-301 airfoil at 12° AoA. Unfortunately the fixed cavity design which enhanced performance at 12°, hurt aerodynamic performance at lower angles of attack. This dilemma led to the development of an AoA adaptive cavity (based on the o19 design) which also proved to enhance aerodynamic performance when located at the AoA-specific point of separation. For the FFA-W3-301 airfoil at 14° AoA, this yielded 19% improvement in L/D. To determine whether the results were extendable to other airfoil geometries, tests run on a NACA0024 airfoil also showed enhanced performance with the AoA adaptive cavity. However, extensive study with different types of airfoils is required to determine applicability of the cavity to enhance performance.

REFERENCES

Box, G. E., & Draper, N. R. (2007). *Response surfaces, mixtures, and ridge analyses.* Wiley.

Cioppa, T. M., & Lucas, T. W. (2007). Efficient nearly orthogonal and space-filling latin hypercubes. *Technometrics, Vol. 49, No. 1*, 45-55.

Fang, K.-T., Li, R., & Sudjianto, A. (2006). *Design and modeling for computer experiments.* 6000 Broken Sound Parkway NW, Suite 300, Boca Raton, FL 33487-2742: Chapman & Hall/CRC, Taylor & Francis Group.

Fang, K.-T., Lin, D. K., Winker, P., & Zhang, Y. (2000). Uniform design: theory and application. *Technometrics, Vol. 42, No. 3*, 237-248.

Friedman, J. H. (1991). Multivariate adaptive regression splines. *The annals of statistics*.

Fuglsang, P., Antoniou, I., Dahl, K. S., & Madsen, H. A. (1998). *Wind tunnel tests of the FFA-W3-241, FFA-W3-301 and NACA 63-430 airfoils.* Roskilde: Riso National Laboratory.

Gregorio, F., & Fraioli, G. (2008). Flow control on a high thickness airfoil by a trapped vortex cavity. *14th International Symposium of Laser Techniques to Fluid Mechanics.* Lisbon, Portugal.

Hickernell, F. J. (1998). A generalized discrepancy and quadrature error bound. *Mathematics of computation, Vol. 67*, 299-322.

Hickernell, F. J. (1998). Lattice rules: how well do they measure up? *Lectire notes in statistics, Vol. 138*, 109-166.

Hoerl, A. E., & Kennard, R. W. (1970). Ridge regression: biased estimation for nonorthogonal problems. *Technometrics, Vol. 12, No. 1*, 55-67.

Jiang, M., & Remotigue, M. (July 1998). GUM-B grid generation code and applications. *Numerical Grid Generation in Computational Field Simulations, Proceedings of the 6th International Conference.* London, England.

Jin, R., Chen, W., & Simpson, T. W. (2000). Comparative studies of metamodeling techniques under multiple modeling criteria. *Structural and multidisciplinary optimization, Vol. 23*, 1-13.

Johnson, M. E., Moore, L. M., & Ylvisaker, D. (1990). Minimax and maximin distance designs. *Journal of statistical planning and inference, Vol. 26, Issue 2*, 131-148.

Kleijnen, J. P. (1986). *Statistical tools for simulation practitioners.* New York, NY: Marcel Dekker, Inc.

Kline, R., & Fogleman, F. (1970). *USA Patent No. 3706430.*

Lawrence, J. (1994). *Introduction to neural networks: design, theory, and applications, sixth edition.* California Scientific Software Press.

Luke, E. A., Tong, X.-L., Wu, J., Tang, L., & Cinnella, P. (2003). *A chemically reacting flow solver for generalized grids.*

Matheron, G. (1963). Principles of geostatistics. *Economic Geology, Vol. 58*, 1246-1266.

McKay, M. D., Beckman, R. J., & Conover, W. J. (2000). A comparison of three methods for selecting values of input variables in the analysis of output from a computer experiment. *Technometrics, Vol. 42, No. 1*, 55-61.

Menter, F. R. (1994). Two equation eddy-viscosity turbulence models for engineering applications. *AIAA Journal, Vol 32, No. 8*, pp. 1598-1605.

Morris, M. D., & Mitchell, T. J. (1995). Exploratory designs for computational experiments. *Journal of statistical planning and inference, Vol. 43, Issue 3*, 381-402.

Mullur, A. A., & Messac, A. (2005). Extended radial basis functions: more flexible and effective metamodeling. *AIAA Journal, Vol. 43, No. 6*, pp. 1306-1315.

Park, J.-S. (1994). Optimal latin-hypercube designs for computer experiments. *Journal of statistical planning and inference, Vol. 39, Issue 1*, 95-111.

Powell, M. (1987). Radial basis functions for multivariable interpolation: a review. *Algorithms for approximation*, p 143-167.

Roache, P. J. (1997). Quantification of uncertainty in computational fluid dynamics. *Annual Review of Fluid Mechanics, 29*, 123-160.

Sacks, J., Welch, W. J., Mitchell, T. J., & Wynn, H. P. (1989). Design and analysis of computer experiments. *Statistical Science, Vol 4, No. 4*, 409-423.

Shewry, M. C., & Wynn, H. P. (1987). Maximum entropy sampling. *Journal of applied statistics, Vol. 14, Issue 2*, 165-170.

Steinberg, D. M., & Lin, D. K. (2006). A construction method for orthogonal latin hypercube designs. *Biometrica, 93, 2*, 279-288.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the royal statistical society, Vol. 58, Issue 1*, 267-288.

Tikhonov, A. N., & Arsenin, V. Y. (1978). Solutions of ill-posed problems. *Mathematics of Computation, Vol. 32, No. 144*, pp. 1320-1322.

Ye, K. Q. (1998). Orthogonal column latin hypercubes and their application in computer experiments. *Journal of american statistical association, Vol. 93, No. 444*, 1430-1439.

Ye, K. Q., William, L., & Sudjianto, A. (2000). Algorithmic construction of optimal symmetric Latin hypercube designs. *Journal of statistical planning and inference, Vol. 90, Issue 1*, 145-159.

APPENDIX A

METAMODEL DEVELOPMENT ON TOY FUNCTIONS

In order to validate the program that was used to build metamodels, two toy functions were considered. Polynomial models up to degree six and radial basis functions with Gaussian, multiquadratic, inverse multiquadratic, and quadratic kernels were trained and the performance metrics of the models were calculated using a sufficiently large test data set for each of these functions.

A.1 First Toy Function

First toy function is a univariate function. It has the form shown in equation A.1 and its graph is shown in figure A.1. The function has moderate fluctuations over most of the domain while the region between 1.0 and 2.0 of the domain shows larger fluctuations.

$$f(x) = \sin(x^2)\cos(x^2) + \sin(2x^2)\cos(2x^2) - x^{1.2} + 3; x \in [0,2] \qquad \text{(A.1)}$$

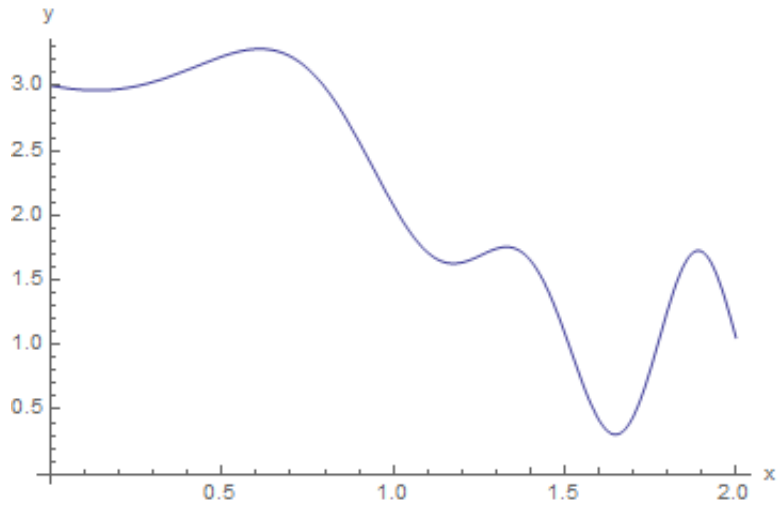


Figure A.1 Graph of function with equation no. A.1

The training data set consisted of 10 uniformly-spaced points in the domain [0, 2]. All of the metamodels were trained using this set. A separate test data set of 50 points

was used to determine the prediction accuracy of the models using the metrics discussed in section 3.4. The performance metrics for the models are shown in figure A.2. A graphical comparison of the metamodels with the original function is presented in figure A.3. Comparison of the cross validation scores didn't give any definitive comparison of the relative performance of each model. Comparison of the R2, RAAE, and RMAE values of all of the models indicated that the radial basis function models have much better prediction accuracy compared to the polynomial models. Among the polynomial models, the second degree polynomial models showed moderate accuracy. With an increase in the degree of the polynomial, more accurate models were obtained. The algorithm used to train the model also affected their accuracy. Polynomial models trained with the OLS and ridge regression showed better accuracy compared to the LASSO regression in this case.

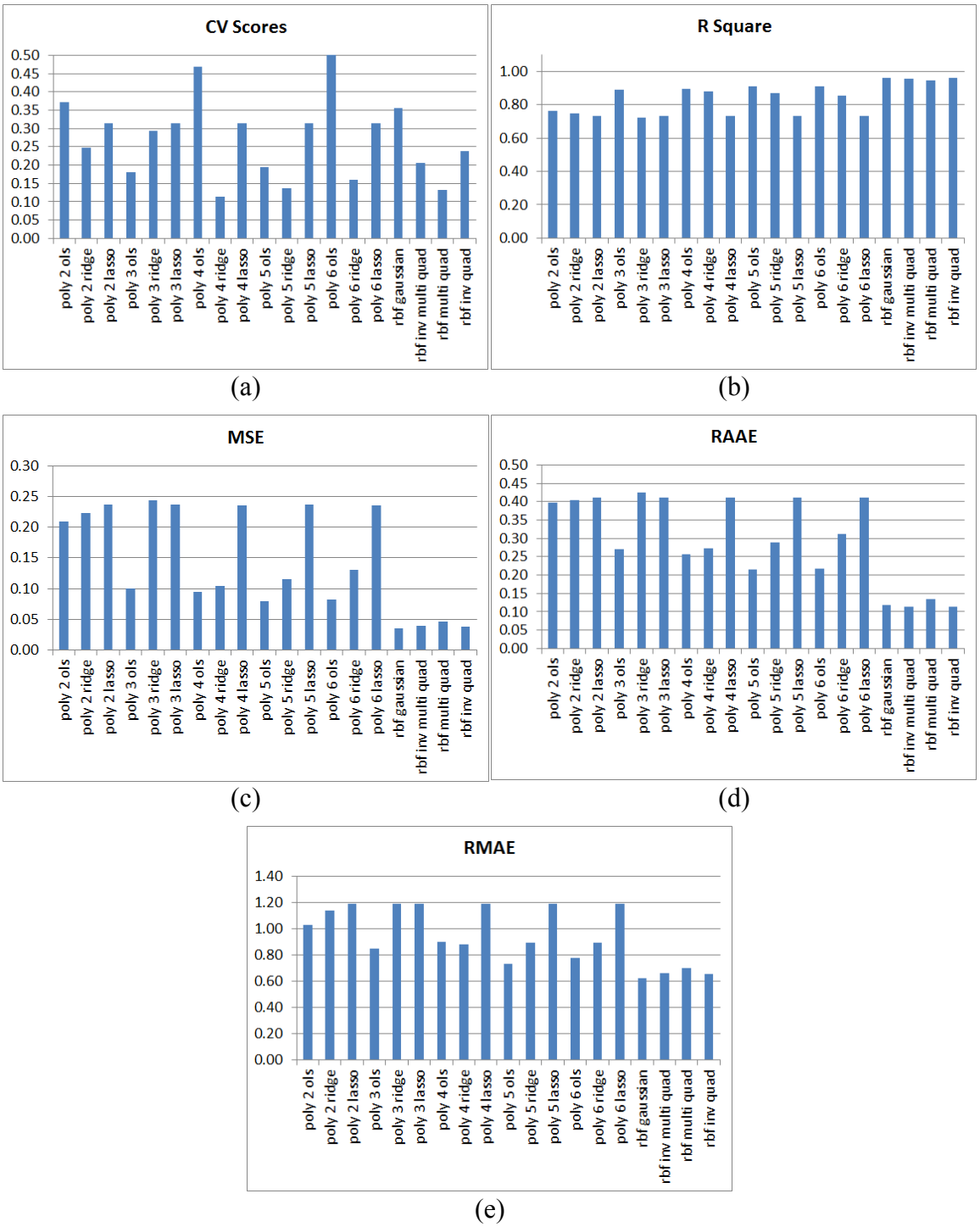
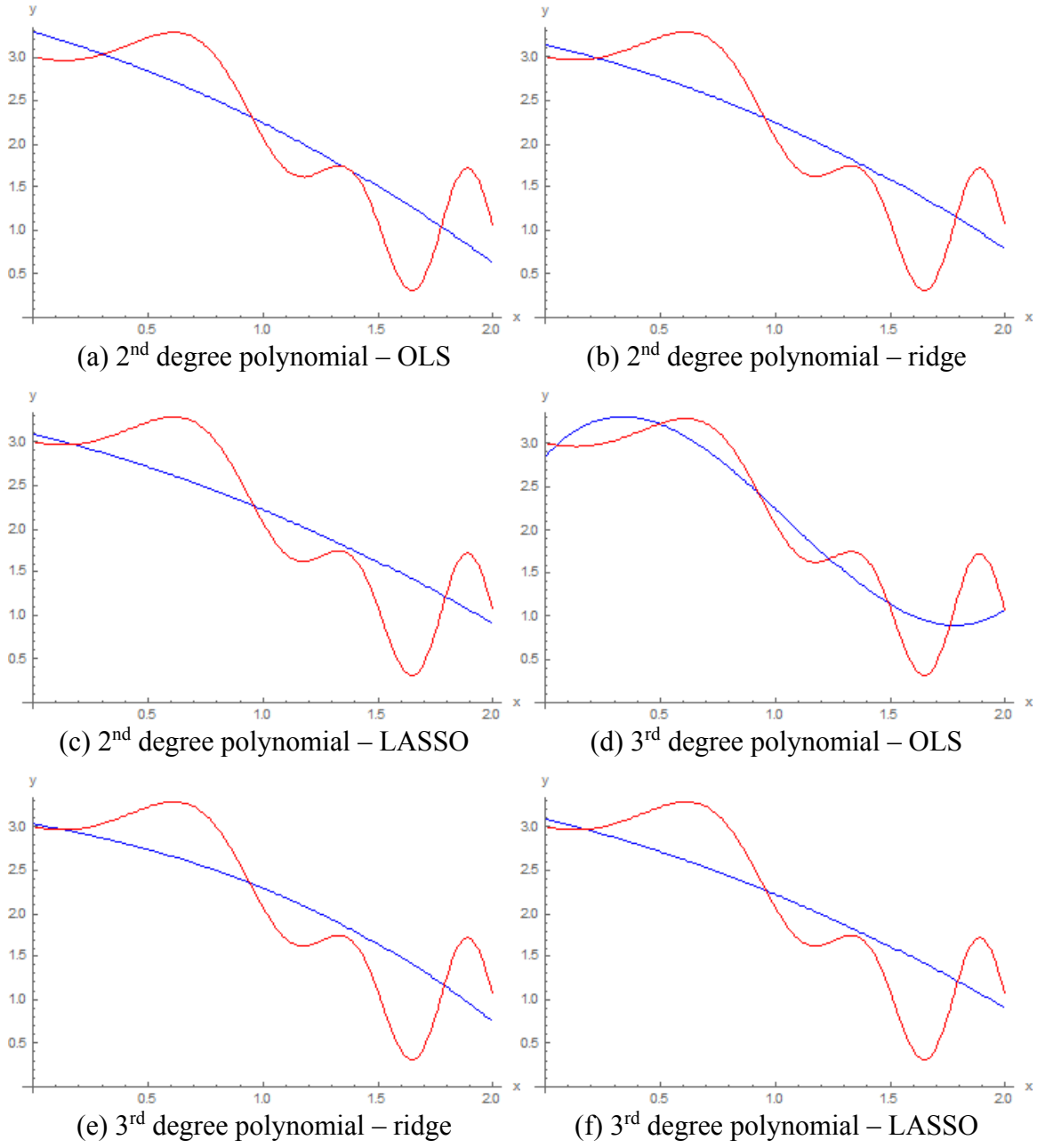


Figure A.2 Comparison of test metrics of the metamodels developed for the first toy function given by equation. A.1.



(a) 2nd degree polynomial – OLS

(b) 2nd degree polynomial – ridge

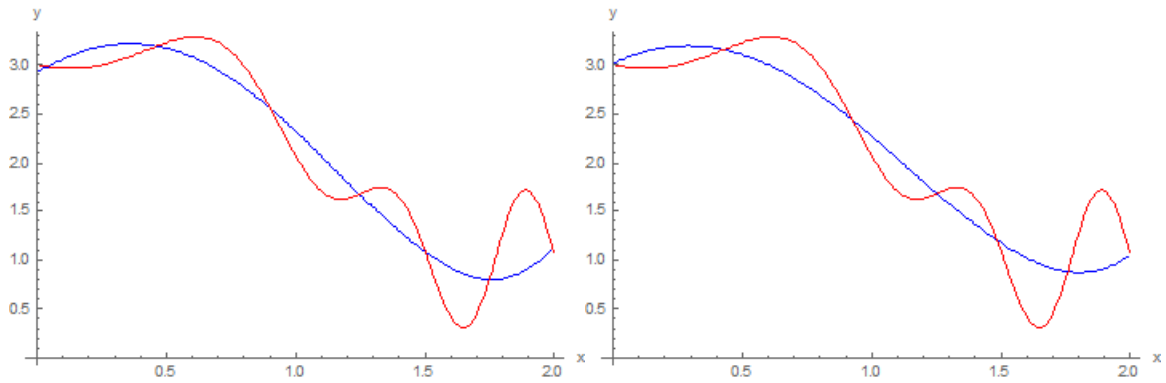
(c) 2nd degree polynomial – LASSO

(d) 3rd degree polynomial – OLS

(e) 3rd degree polynomial – ridge

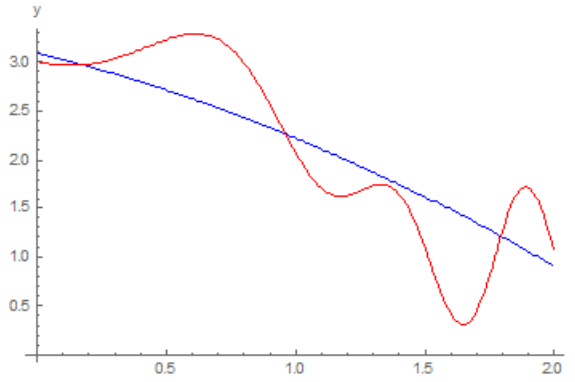
(f) 3rd degree polynomial – LASSO

Figure A.3 Graphical comparison of toy function 1 (equation A.1) and its metamodels

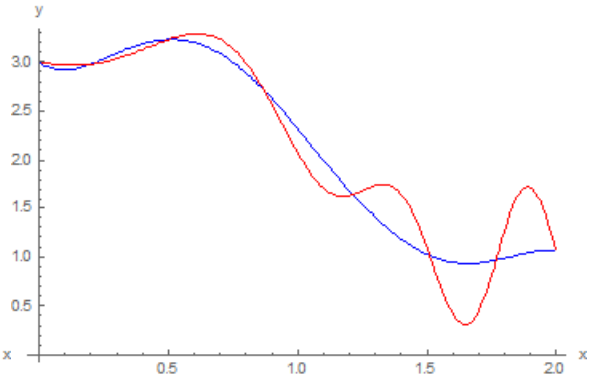


(g) 4th degree polynomial – OLS

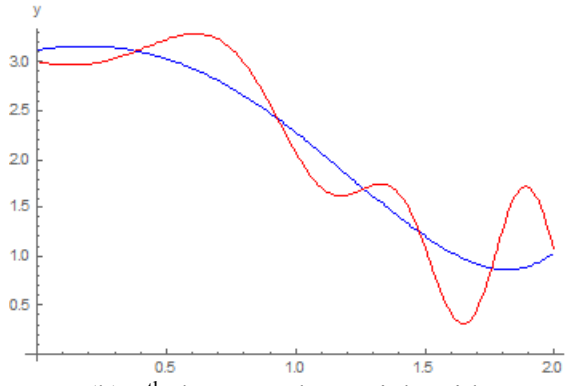
(h) 4th degree polynomial – ridge



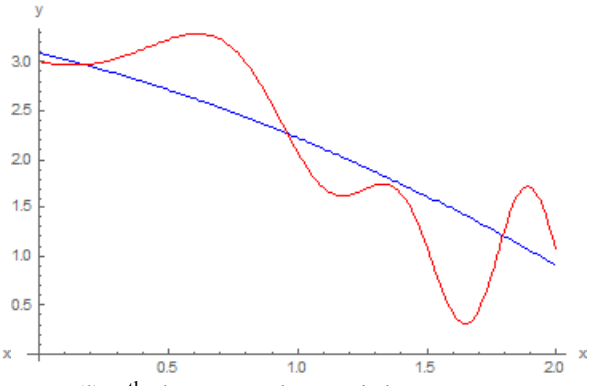
(i) 4th degree polynomial – LASSO



(j) 5th degree polynomial – OLS

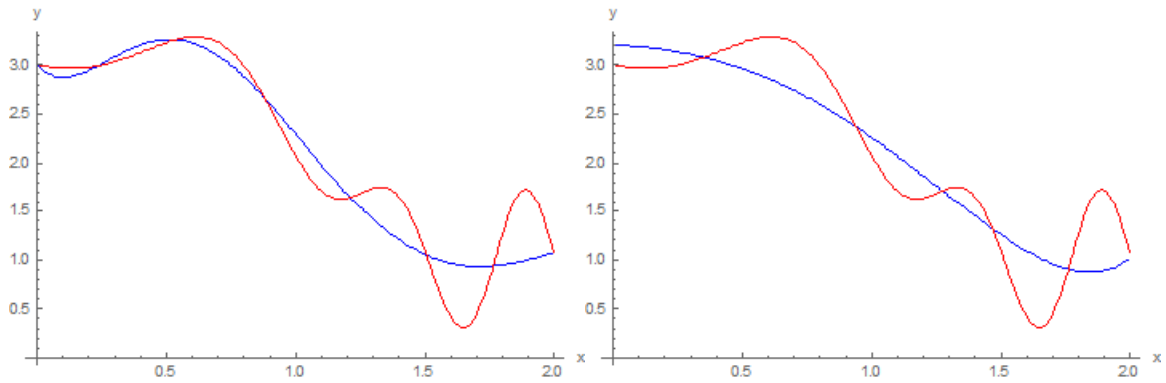


(k) 5th degree polynomial – ridge



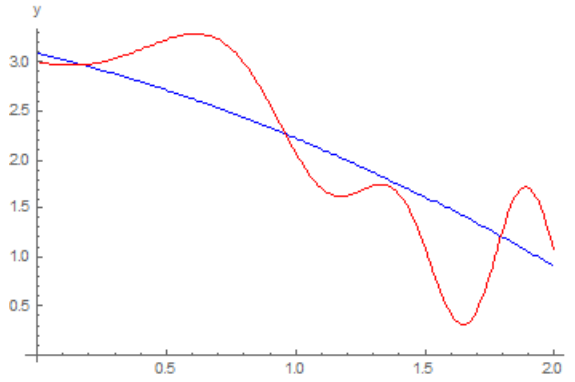
(l) 5th degree polynomial – LASSO

Figure A.3 (continued)

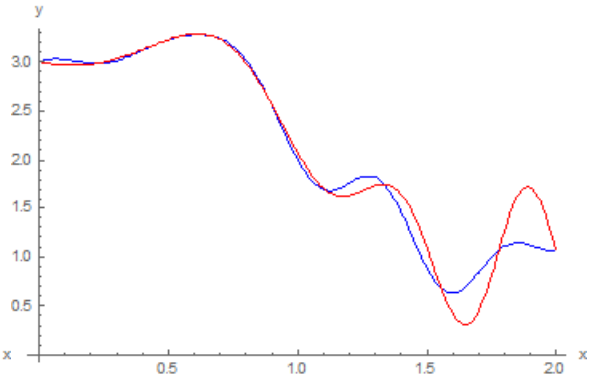


(m) 6th degree polynomial – OLS

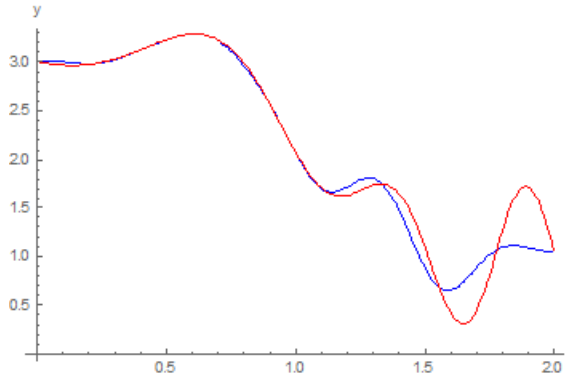
(n) 6th degree polynomial – ridge



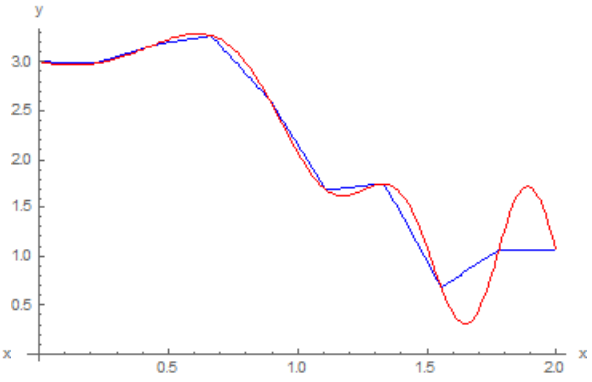
(o) 6th degree polynomial – LASSO



(p) RBF – Gaussian

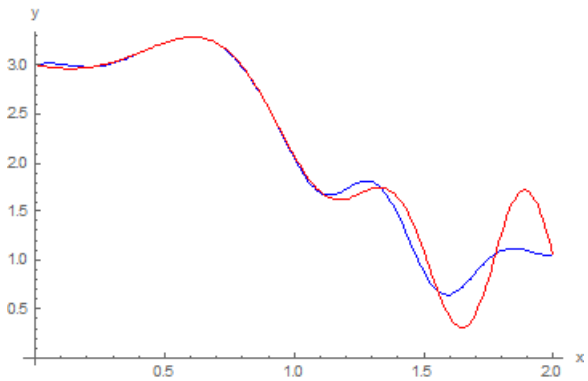


(q) RBF – Inverse Multiquadratic



(r) RBF – Multiquadratic

Figure A.3 (continued)



(s) RBF – Inverse Quadratic

Figure A.3 (continued)

A.2 Second Toy Function

The second toy function is a bivariate function of the form given by equation A.2. The domain of the function is $x_1, x_2 \in [0.5,3.5]$.

$$f(x_1, x_2) = 2 + 4x_1 + 4x_2 - {x_1}^2 - {x_2}^2 + 2\sin(2x_1)\sin(2x_2) \qquad \text{(A.2)}$$

The graph of this function is as shown in figure A.4. The function was modeled using a symmetric Latin hypercube design of 20 points generated using the columnwise-pairwise exchange algorithm (Ye, William, & Sudjianto, 2000) to achieve the lowest discrepancy among the explored designs. The trained metamodels were tested against a 100 point test data set with a symmetric Latin hypercube structure that fills the design space as uniformly as possible. The test metrics are shown in figure A.5. The surface plots of the metamodels are shown in figure A.6.

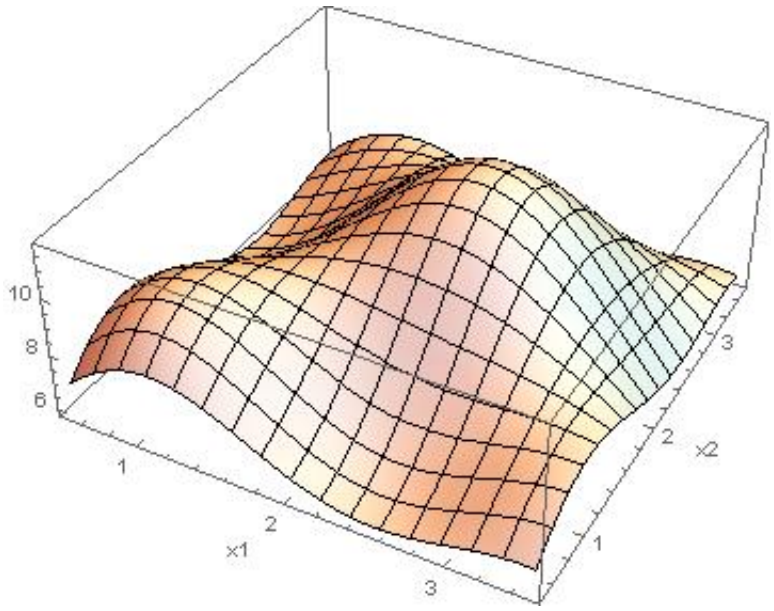
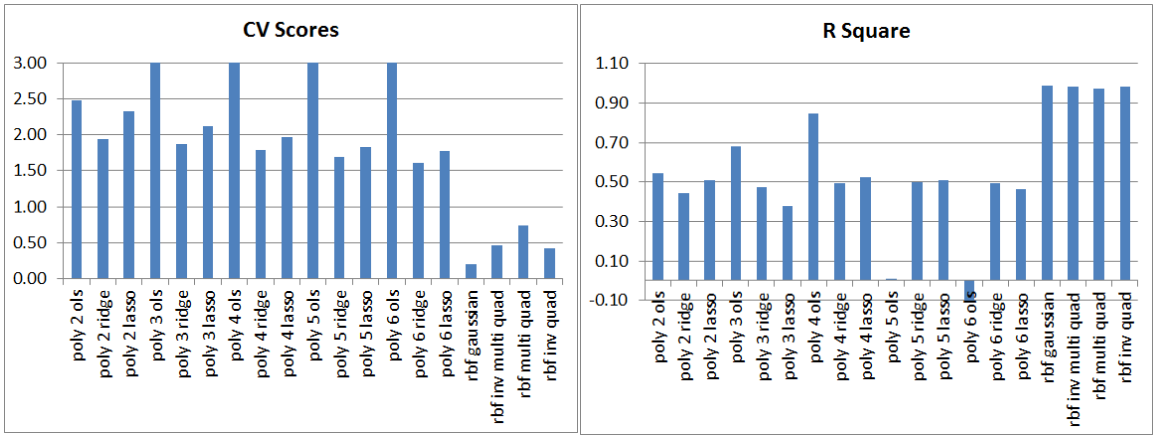
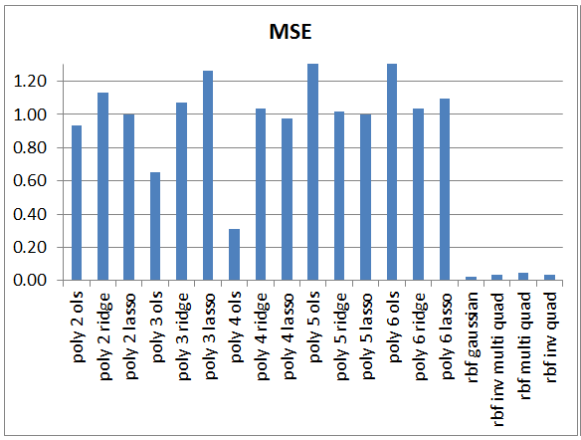


Figure A.4 Surface plot of the second toy function given by equation A.2

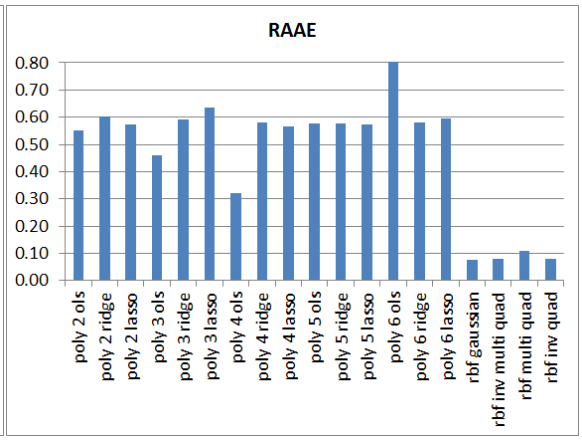


(a) (b)

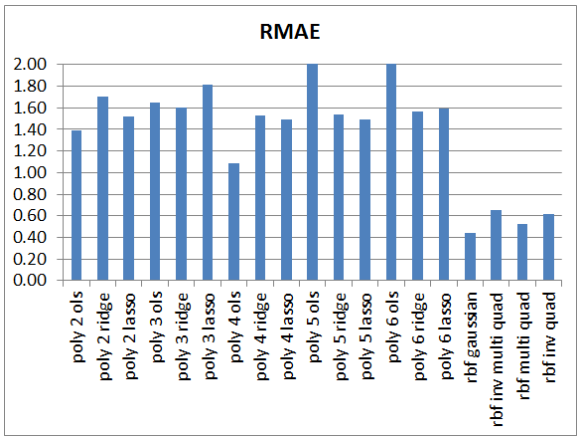
Figure A.5 Comparison of test metrics of the metamodels developed for the second toy function given by equation A.2



(c)

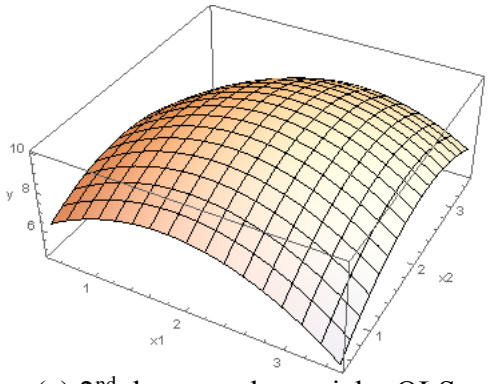


(d)

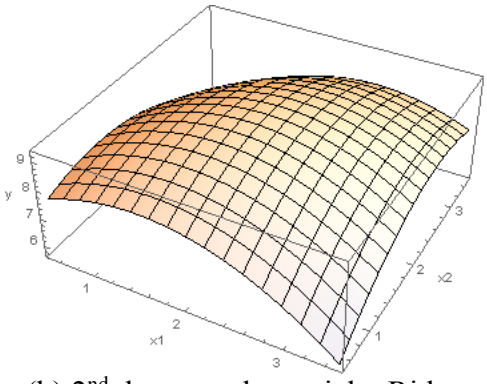


(e)

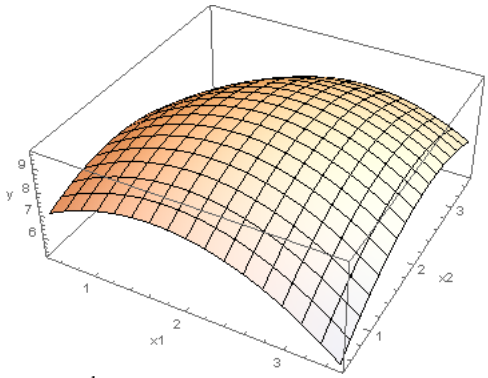
Figure A.5 (continued)



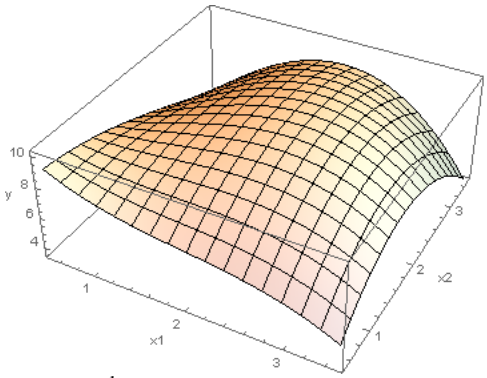
(a) 2nd degree polynomial – OLS



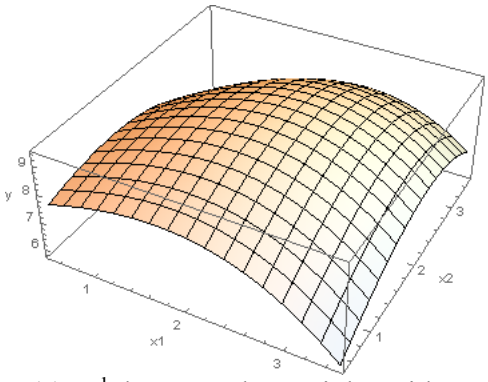
(b) 2nd degree polynomial – Ridge



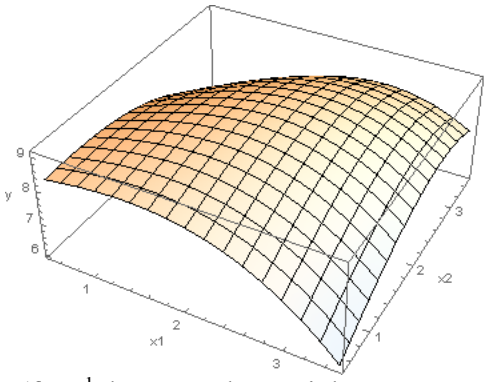
(c) 2nd degree polynomial – LASSO



(d) 3rd degree polynomial – OLS

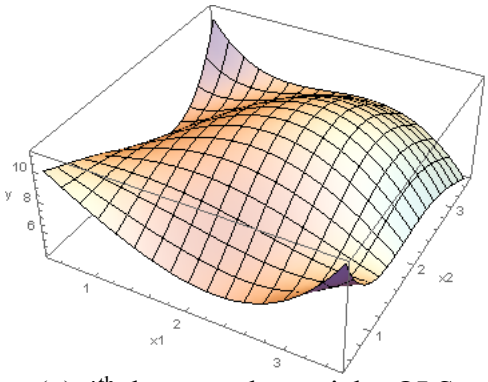


(e) 3rd degree polynomial – Ridge

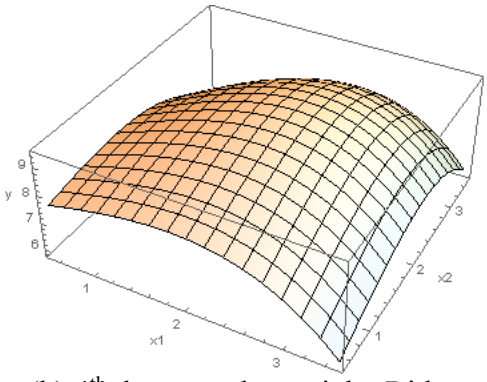


(f) 3rd degree polynomial – LASSO

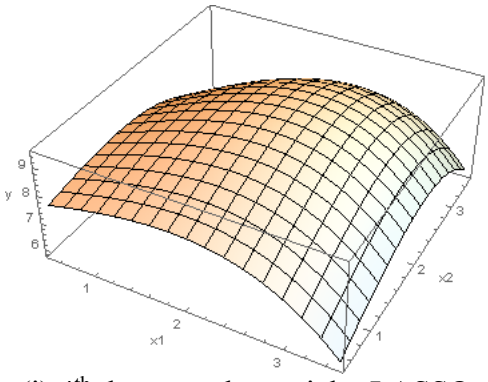
Figure A.6 Plots of the metamodels developed for the second toy function given by equation A.2.



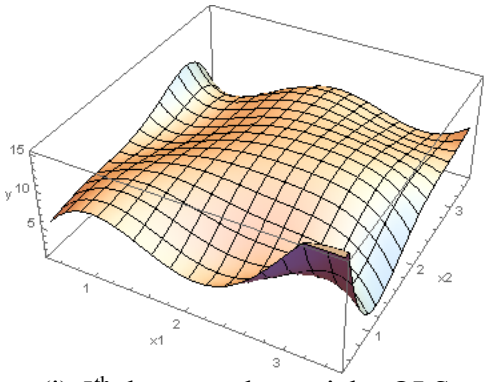
(g) 4th degree polynomial – OLS



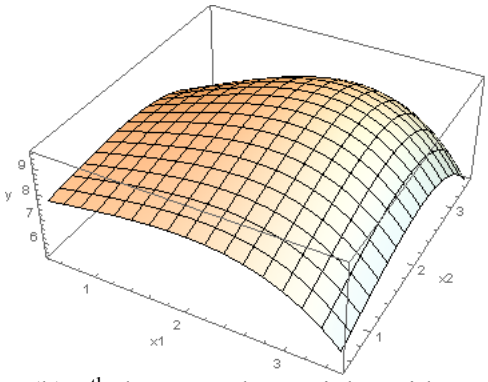
(h) 4th degree polynomial – Ridge



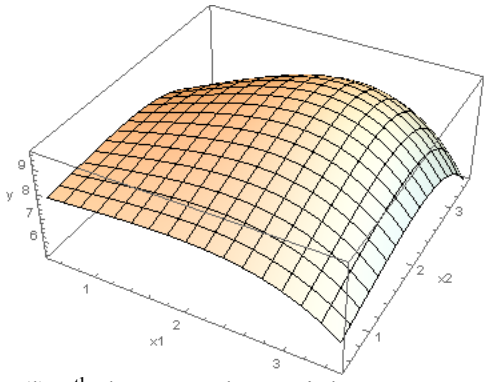
(i) 4th degree polynomial – LASSO



(j) 5th degree polynomial – OLS

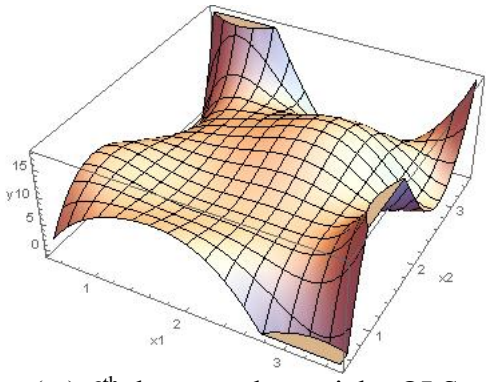


(k) 5th degree polynomial – Ridge

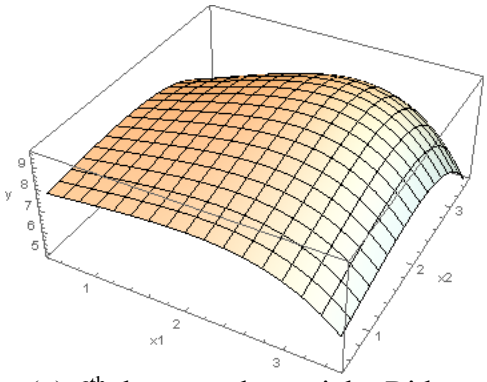


(l) 5th degree polynomial – LASSO

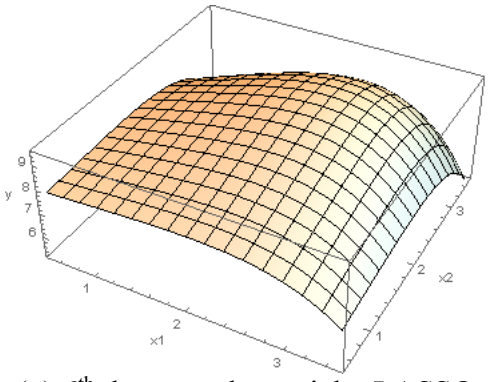
Figure A.6 (continued)



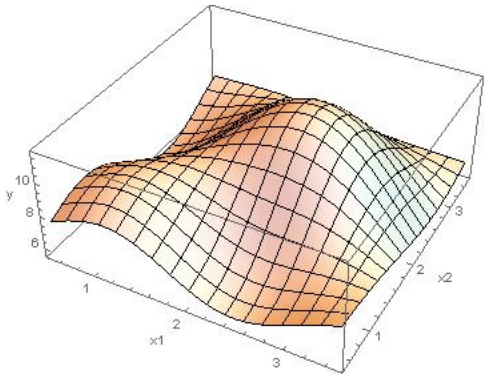
(m) 6th degree polynomial – OLS



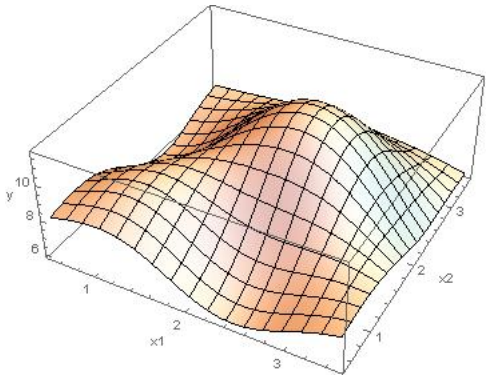
(n) 6th degree polynomial – Ridge



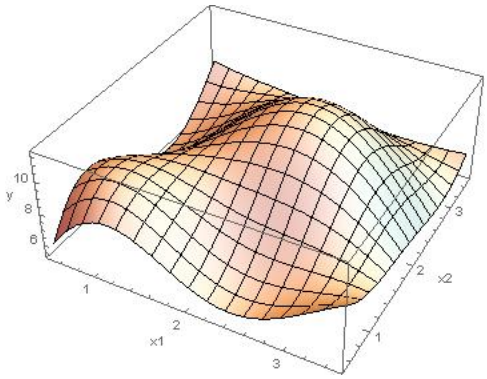
(o) 6th degree polynomial – LASSO



(p) RBF – Gaussian

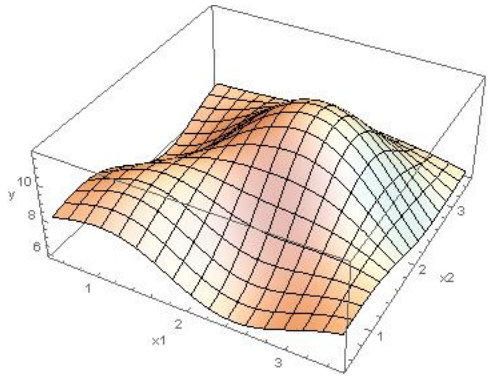


(q) RBF – Inverse Multiquadratic



(r) RBF – Multiquadratic

Figure A.6 (continued)



(s) RBF – Inverse quadratic

Figure A.6 (continued)

The cross validation scores, R^2, RAAE, and RMAE indicated that the the radial basis functions had much better accuracy compared to the polynomial models. Among the polynomial models, the 4th-degree polynomial models trained with the OLS regression showed better accuracy compared to the other polynomial models.

APPENDIX B

EXPERIMENTAL DESIGN FOR THE FFA-W3-301 AIRFOIL CAVITY SHAPE OPTIMIZATION

The design points used to train the metamodels of performance of the FFA-W3-301 airfoil with cavity are listed in table B.1. It is a symmetric Latin hypercube design optimized by performing columnwise-pairwise exchange algorithm with centered discrepancy as the optimization objective in order to achieve better space-filling property.

Table B.1 Experimental design for training metamodels of performance of the FFA-W3-301 airfoil with cavity

No.	x_1	x_2	x_3	x_4	x_5	x_6	x_7
1	0.5485	38.8636	0.0773	42.9545	0.0788	0.7500	0.0000
2	0.5924	42.2727	0.0932	43.6364	0.0992	0.7242	0.0179
3	0.5636	64.7727	0.0970	45.6818	0.0576	0.6652	0.0170
4	0.5530	23.1818	0.0795	40.9091	0.0674	0.6545	0.0155
5	0.5667	48.4091	0.0758	62.0455	0.0970	0.7470	0.0133
6	0.5773	61.3636	0.0629	25.9091	0.0841	0.6667	0.0173
7	0.5606	37.5000	0.0606	40.2273	0.0955	0.6803	0.0194
8	0.5258	62.7273	0.0811	38.1818	0.0553	0.6970	0.0136
9	0.5576	22.5000	0.0985	66.1364	0.0879	0.7045	0.0073
10	0.5288	54.5455	0.0538	31.3636	0.0689	0.6636	0.0070
11	0.5121	60.6818	0.0621	56.5909	0.0682	0.7318	0.0164
12	0.5833	65.4545	0.0780	55.9091	0.0705	0.6818	3.0303e-04
13	0.5697	47.0455	0.0667	22.5000	0.0985	0.7015	0.0012
14	0.5348	53.1818	0.0583	61.3636	0.0962	0.6939	0.0142
15	0.5788	30.6818	0.0545	53.8636	0.0742	0.7258	0.0176
16	0.5591	34.0909	0.0523	24.5455	0.0780	0.6879	0.0124
17	0.5455	56.5909	0.0652	34.7727	0.0500	0.7439	0.0091
18	0.5864	25.9091	0.0674	66.8182	0.0523	0.6697	0.0106
19	0.5242	66.1364	0.0682	51.1364	0.0909	0.6864	0.0061
20	0.5894	50.4545	0.0841	32.7273	0.0614	0.7424	0.0015
21	0.5152	34.7727	0.0697	41.5909	0.0606	0.7227	0.0055
22	0.5045	57.2727	0.0902	42.2727	0.0735	0.6909	0.0167
23	0.5182	45.6818	0.0500	64.7727	0.0833	0.6712	0.0121
24	0.5439	43.6364	0.0614	27.2727	0.0902	0.7212	0.0039
25	0.5970	63.4091	0.0924	60.6818	0.0773	0.7379	0.0097
26	0.5682	49.0909	0.0765	30.0000	0.0644	0.6848	0.0118
27	0.5061	49.7727	0.0788	59.3182	0.0803	0.7076	0.0042
28	0.5985	51.8182	0.0947	39.5455	0.0871	0.6515	0.0112
29	0.5727	57.9545	0.0636	63.4091	0.0652	0.6591	0.0018

Table B.1 (continued)

No.	x_1	x_2	x_3	x_4	x_5	x_6	x_7
30	0.5621	30.0000	0.0644	35.4545	0.0583	0.7394	0.0148
31	0.5091	27.9545	0.0939	52.5000	0.0636	0.6833	0.0115
32	0.5803	58.6364	0.0508	53.1818	0.0568	0.7273	0.0191
33	0.6000	36.1364	0.0591	57.9545	0.0561	0.6894	0.0048
34	0.5500	45.0000	0.0750	45.0000	0.0750	0.7000	0.0100
35	0.5000	53.8636	0.0909	32.0455	0.0939	0.7106	0.0152
36	0.5197	31.3636	0.0992	36.8182	0.0932	0.6727	9.0909e-04
37	0.5909	62.0455	0.0561	37.5000	0.0864	0.7167	0.0085
38	0.5379	60.0000	0.0856	54.5455	0.0917	0.6606	0.0052
39	0.5273	32.0455	0.0864	26.5909	0.0848	0.7409	0.0182
40	0.5015	38.1818	0.0553	50.4545	0.0629	0.7485	0.0088
41	0.5939	40.2273	0.0712	30.6818	0.0697	0.6924	0.0158
42	0.5318	40.9091	0.0735	60.0000	0.0856	0.7152	0.0082
43	0.5030	26.5909	0.0576	29.3182	0.0727	0.6621	0.0103
44	0.5561	46.3636	0.0886	62.7273	0.0598	0.6788	0.0161
45	0.5818	44.3182	0.1000	25.2273	0.0667	0.7288	0.0079
46	0.5955	32.7273	0.0598	47.7273	0.0765	0.7091	0.0033
47	0.5848	55.2273	0.0803	48.4091	0.0894	0.6773	0.0145
48	0.5106	39.5455	0.0659	57.2727	0.0886	0.6576	0.0185
49	0.5758	23.8636	0.0818	38.8636	0.0591	0.7136	0.0139
50	0.5136	64.0909	0.0826	23.1818	0.0977	0.7303	0.0094
51	0.5545	33.4091	0.0848	55.2273	0.1000	0.6561	0.0109
52	0.5409	55.9091	0.0977	65.4545	0.0720	0.7121	0.0076
53	0.5212	59.3182	0.0955	36.1364	0.0758	0.6742	0.0024
54	0.5652	36.8182	0.0917	28.6364	0.0538	0.7061	0.0058
55	0.5303	42.9545	0.0833	67.5000	0.0515	0.6985	0.0188
56	0.5167	24.5455	0.0720	34.0909	0.0795	0.7182	0.0197
57	0.5879	29.3182	0.0879	33.4091	0.0818	0.6682	0.0036
58	0.5712	35.4545	0.0962	58.6364	0.0811	0.7364	0.0130
59	0.5424	67.5000	0.0515	23.8636	0.0621	0.6955	0.0127
60	0.5742	27.2727	0.0689	51.8182	0.0947	0.7030	0.0064
61	0.5394	52.5000	0.0894	49.7727	0.0545	0.7197	6.06E-04
62	0.5227	28.6364	0.0871	64.0909	0.0659	0.7333	0.0027
63	0.5333	41.5909	0.0742	27.9545	0.0530	0.6530	0.0067
64	0.5470	66.8182	0.0705	49.0909	0.0826	0.7455	0.0045
65	0.5364	25.2273	0.0530	44.3182	0.0924	0.7348	0.0030
66	0.5076	47.7273	0.0568	46.3636	0.0508	0.6758	0.0021
67	0.5515	51.1364	0.0727	47.0455	0.0712	0.6500	0.0200