

CFD BASED AIRFOIL SHAPE OPTIMIZATION FOR AERODYNAMIC DRAG
REDUCTION

by

Mohammed Taha Shafiq Khot

A Thesis Presented to the Faculty of the American University of Sharjah
College of Engineering
in Partial Fulfillment of
the Requirements for the Degree of

Master of Science in
Mechanical Engineering

Sharjah, United Arab Emirates

May 2012

Approval Signatures

We, the undersigned, approve the Master's Thesis of Mohammed Taha Shafiq Khot.

Thesis Title: CFD Based Airfoil Shape Optimization for Aerodynamic Drag Reduction


Signature

Date of Signature



20/06/2012

Dr. Ali Jhemi
Assistant Professor
Department of Mechanical Engineering
Thesis Advisor




27-June-12

Dr. Mohammad Amin Al Jarrah
Professor
Department of Mechanical Engineering
Thesis Committee Member



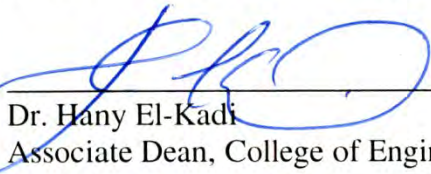
25/6/2012

Dr. Essam Wahba
Associate Professor
Department of Mechanical Engineering
Thesis Committee Member



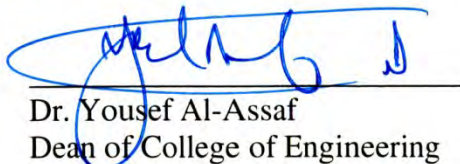
27-June-12

Dr. Mohammad Amin Al Jarrah
Department Head
Department of Mechanical Engineering



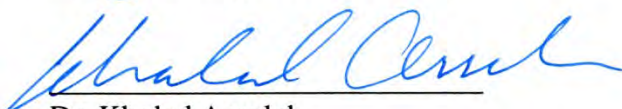
27/6/2012

Dr. Hany El-Kadi
Associate Dean, College of Engineering



28/6/12

Dr. Yousef Al-Assaf
Dean of College of Engineering



03/07/2012

Dr. Khaled Assaleh
Director of Graduate Studies

Abstract

Commercial airplanes generally follow specific flight profiles consisting of take-off, climb, cruise, descend and landing. These flight profiles essentially change the freestream conditions in which the aircrafts operate. Furthermore, over the course of the flight, the required lift force changes as the fuel gets consumed. The conventional fixed wing designs account for these requirements by catering to multiple but fixed design points which, however, compromise the overall flight performance. Employing adaptive wing technology allows to fully exploring the aerodynamic flow potential at each point of the flight envelope. The objective of this research is to develop an aerodynamic optimization framework for optimizing a baseline airfoil shape at specific off-design operating points within a typical transonic flight envelope. The objective function is the lift-moment constrained drag minimization problem. B-spline curve fitting is used to parameterize the airfoil geometry and the control points along with the angle of attack are used as the design variables for the optimization process. An iterative response surface optimization methodology is employed for carrying out the shape optimization process. Design of Experiments (DoE) using the Latin Hypercube Sampling algorithm is used to construct the response surface model. This model is then optimized using the SQP technique. The various parameters that gauge the aerodynamic performance (lift, drag and moment coefficients) are obtained using CFD simulations. GridPro is used as the meshing tool to generate the flow mesh, and the CFD simulation is performed using ANSYS-FLUENT. The parameterization, design of experiments, and the response surface model optimization are performed using MATLAB. RAE 2822 design study is carried out to validate the optimization algorithm developed. The adaptive airfoil concept is demonstrated using a Boeing-737 classic airfoil at three steady flight operating points that lie within a typical aircraft flight envelope. The aerodynamic performance of the adaptive airfoil is then compared to that of the baseline airfoil.

Search Terms—Aerodynamics, CFD, optimization, design of experiments, response surface modeling.

Table of Contents

Abstract	4
List of Figures	9
List of Tables	11
Nomenclature	12
Acknowledgements	13
1. INTRODUCTION	15
1.1 Motivation	16
1.2 Historical perspective	17
1.3 Problem Statement	22
1.4 Review of Literature	24
1.5 Thesis Objectives	26
2. OPTIMIZATION AND CFD	28
2.1 Introduction to Optimization	28
2.1.1 Mathematical Formulation	29
2.1.2 Characteristics of Optimization Algorithms	29
2.2 Sequential Quadratic Programming	30
2.2.1 The SQP Technique	30
2.2.2 Solution Procedure	31
2.2.3 Updating the Hessian Matrix	31
2.3 Design Variables	32
2.3.1 B-Spline Curves Introduction	32
2.3.2 B-Spline Formulation	33
2.3.3 Airfoil parameterization using B-Spline curves	33
2.3.4 Design Variables Vector	38
2.4 Objective Functions	38
2.4.1 Inverse design	38

2.4.2 Maximizing Aerodynamic Efficiency.....	39
2.4.3 Lift-Constrained Drag Minimization	39
2.5 Flow Equations.....	40
2.5.1 The Mass Conservation Equation	41
2.5.2 Momentum Conservation Equations.....	42
2.5.3 Viscous Energy Equation.....	42
2.5.4 Turbulence Model	43
2.6 Computational Fluid Dynamics	48
2.6.1 Discretization	49
2.6.2 CFD based Shape Optimization.....	51
3. ITERATIVE RESPONSE SURFACE BASED OPTIMIZATION	52
3.1 Introduction	52
3.2 Response Surface Optimization	53
3.2.1 Design Space.....	55
3.2.2 Design of Experiments.....	56
3.2.3 Constructing the RSM	61
3.2.4 Optimizing the RSM	65
3.2.5 System Response	65
3.3 Iterative Improvement of the RSM	66
3.3.1 Algorithm Flow Chart.....	67
3.4 MATLAB Implementation of the Iterative RSO	70
3.4.1 MATLAB Optimization Toolbox	74
3.4.2 MATLAB Global Optimization Toolbox	77
3.5 Meshing using GridPro	79
3.5.1 Surface Specifications.....	79
3.5.2 Block Topology	81
3.5.3 Run schedule.....	84

3.5.4 Wall Clustering	84
3.6 CFD Solver—ANSYS FLUENT	86
3.6.1 Pressure-Based Solver	87
3.6.2 General Scalar Transport Equation.....	87
3.6.3 Spatial Discretization	89
3.6.4 Computing Forces and Moments	90
3.6.5 ANSYS FLUENT Setup and Run.....	91
3.6.6 Flow Solver Validation	93
3.7 Simulation Time.....	94
4. RESULTS AND DISCUSSIONS	96
4.1 Overview	96
4.2 Case Study#1: Optimization of the RAE 2822 Airfoil	96
4.3 Adaptive Airfoil Design based on the Boeing-737C airfoil.....	101
4.3.1 Operational Envelope.....	102
4.3.2 C_L requirements across the envelope	103
4.3.3 Airfoil Geometry.....	104
4.3.4 CASE-A: Mach 0.74 at 9083.3 [m] (29,800[ft])	105
4.3.5 CASE-B: Mach 0.65 at 9083.3 [m] (29,800[ft]).....	109
4.3.6 CASE-C: Mach 0.65 at 6827.0 [m] (22,400[ft]).....	113
4.4 Performance Comparison of Baseline Airfoil Relative to Optimized Airfoil..	117
4.5 Adaptive Airfoils.....	118
5. CONCLUSION AND FUTURE WORK.....	120
5.1 Summary and Conclusion	120
5.2 Future Work	123
REFERENCES	125
Appendix A GridPro Topology Input Language (TIL) File (FINAL3.FRA).....	128
Appendix B ANSYS-FLUENT JOURNAL FILE (FLUENT_RUN.JOU).....	136

VITA..... 138

List of Figures

Figure 1 AFTI-F-111 Mission Adaptive Wing (MAW) in Flight	18
Figure 2 Optimum wing cambers at different flight conditions	19
Figure 3 Variable leading and trailing edge wing design	19
Figure 4 Lewis et al variable camber wing command system	20
Figure 5 Artist's rendering of NASA's Morphing Airplane.....	21
Figure 6 Reparameterization.....	35
Figure 7 B-Spline parameterization of RAE 2822 airfoil.....	37
Figure 8 Continuous Domain and Discrete Domain.....	49
Figure 9 The Computational Spatial Grid.....	49
Figure 10 Typical finite volume cell.....	50
Figure 11 General RSO procedure.....	55
Figure 12 Three-variable, ten-point Latin hypercube sampling plan shown in three dimensions (top left), along with its two-dimensional projections.....	58
Figure 13 Lower Limit, Upper limit and Baseline Control Points and Airfoil shapes	60
Figure 14 Iterative RSO process flow chart.....	69
Figure 15 MATLAB implementation of the Iterative RSO method.....	73
Figure 16 GridPro surfaces	80
Figure 17 Topology layout for the airfoil and the far-field	82
Figure 18 Topology layout in the wake region of the airfoil.....	83
Figure 19 Topology around the airfoil surface	83
Figure 20 Mesh—full display	85
Figure 21 Mesh—full and close up view	85
Figure 22 CFD verification results	94
Figure 23 B-spline parameterization of the RAE 2822 airfoil.....	97
Figure 24 Range of the B-spline control points (Design variables).....	97
Figure 25 RAE 2822 Baseline and Optimized airfoil shapes C_p distributions	100
Figure 26 Plot comparing the actual and predicted function values.....	101
Figure 27 Typical aircraft flight envelope	102
Figure 28 Discretized points in the flight envelope.	103
Figure 29 Boeing 737 airfoil (baseline).....	104
Figure 30 B-Spline parameterization of the Boeing 737 airfoil (expanded)	104
Figure 31 Range of the Boeing-737 airfoil design variables	105

Figure 32 CASE-A Baseline and Optimized airfoil shapes and their corresponding surface C_p distributions.....	108
Figure 33 Plot comparing the actual and the predicted objective function values (CASE-A)	109
Figure 34 CASE-B Baseline and Optimized airfoil shapes and their corresponding surface C_p distributions.....	112
Figure 35 Plot comparing the actual and the predicted objective function values (CASE-B).....	113
Figure 36 Plot comparing the actual and the predicted objective function values (CASE-C).....	116
Figure 37 Plot comparing the actual and the predicted objective function values (CASE-C).....	117
Figure 38 Baseline and Optimized Airfoil shapes for Cases A, B, and C	119
Figure 39 Baseline and Optimized Airfoil shapes for Cases A, B, and C (Expanded View).....	119

List of Tables

Table 1 Model Constants	48
Table 2 Baseline and Upper and Lower limit Values of the Design Variables	98
Table 3 Freestream conditions for the design operating condition	98
Table 4 Baseline and Optimized performance at off-design operating point	99
Table 5 Baseline and Optimized values of the Design Variables	100
Table 6 Baseline and Upper and Lower limit Values of the Design Variables	105
Table 7 Freestream conditions for the off-design operating point A	106
Table 8 Baseline and Optimized values of the Design Variables	106
Table 9 Baseline and Optimized performance at off-design operating point	107
Table 10 Freestream conditions for the off-design operating point B	110
Table 11 Baseline and Optimized values of the Design Variables	110
Table 12 Baseline and Optimized performance at off-design operating point B	111
Table 13 Freestream conditions for the off-design operating point C	114
Table 14 Baseline and Optimized values of the Design Variables C	114
Table 15 Baseline and Optimized performance at off-design operating point C	115
Table 16 Performance Comparison Analysis	118

Nomenclature

f – Objective function

g_i – Constraints

\mathbf{X} — Vector of design variables

L — Lagrangian formulation

λ — Lagrange Multipliers

H — Hessian of the Lagrange function

t_j — B-Spline curve Parameter

$P_i^{x,y}$ — B-Spline Control point coordinates

U_i — B-Spline Knot Vector

C_L, C_D, C_M — Lift, moment and drag coefficients.

J — Aerodynamic optimization objective function

CFD — Computational Fluid Dynamics

\mathbf{V} — Flow velocity vector

ρ — Fluid density

τ_{ij} — Reynolds Stresses

ν — Kinematic Viscosity

\mathbf{X}_{DOE} — Design of Experiments plan

$\hat{\mathbf{y}} = \hat{f}(\mathbf{x}, \boldsymbol{\beta})$ — Response Surface Model

$\boldsymbol{\beta}$ — Polynomial regression coefficients

n_t — Number of regression coefficients.

Φ — Vandermonde matrix of design variables

N_s, N_{var} — Number of samples and variables

σ_a — Adjusted Root Mean Square value

R_{adj}^2 — Coefficient of multiple determination

Acknowledgements

I would like to begin with thanking Almighty God for being my guidance of the journey of life. He is the Creator and Sustainer of the all perfect nature which has been a major source of inspiration for many of the engineering solutions that exist today.

I would like to extend my sincerest thanks and appreciation to Dr. Ali Jhemi. It has been an honor and a privilege to be his student. He is a great mentor whose expert opinions and guidance have been central to the successful completion of this research. Working with him has been both, an enjoyable as well as a great learning experience. Throughout the duration of my research work, Dr. Jhemi has been patient and cordial, which encouraged me to work even harder and with greater determination.

I would especially like to thank Dr. Essam Wahba for helping me understand and clarifying a lot of doubts with respect to CFD. His expert comments and opinions on my simulation results have helped me save a lot of time and do things the right way with confidence. Also I would like to thank Dr. Mohammed Amin Al Jarrah, head of the Mechanical Engineering department, for being a member of my graduate thesis committee and gauging my research work. His experience in the area of aerodynamics has been valuable in defining the scope of my research.

I am very grateful to Dr. Armando Vavalle of Rolls Royce Plc, for his expert opinions on my results which were crucial for my research. He has been very forthcoming and helpful, and his recommendations have been very valuable in making my research work even better. A special appreciation and thanks to Mr. Samuel Ebenezer of GridPro, for not only providing me with the software but even tons of information and guidance relating to it.

Finally, I would like to express my deepest gratitude to everyone in my family, especially my loving parents who have been very kind and understanding throughout. I owe a lot of respect to them for their fabulous upbringing, taking great care of me all the while, supporting and motivating me to work harder to get to this point in life. And so, I dedicate this research to them. Also, I would like to thank all my friends for the great support and for the good times we have.

“Nature makes the ultimate decision as to whether a flow will be laminar or turbulent. There is a general principle in nature, that a system, when left to itself, will always move toward its state of maximum disorder. To bring order to a system, we generally have to exert some work on the system or expend energy in some manner.”

—John Anderson Jr.

(In *Fundamentals of Aerodynamics*)

1. INTRODUCTION

Conventional non-renewable sources of energy, primarily fossil fuels, have essentially been the key drivers of almost all modern economies. For over a century, locomotion and hence the transportation industry has been one of the largest consumer of oil, which has over time, led to a steady increase in demand for the same. As new economies emerge and the existing well established ones continue to sustain and diversify, as trade and business becomes more and more global, the transportation sector will continue to play a major role in order to meet the demands of large scale globalization. This translates into increased demand for oil which as a consequence of reduced supply, leads to increase in fuel costs and thus higher operational costs.

The aviation industry in the last few decades has witnessed a huge surge, attributed primarily to the growth of new and emerging economies in Asia. As cited in a forecast report by the Boeing Company, passenger air traffic rose by almost 8 percent in 2010 after a slight slump in 2009. The world air cargo traffic saw an estimated 24 percent growth in 2010. The global gross domestic product (GDP) is projected to grow at an average of 3.3 percent per year for the next 20 years. Reflecting this economic growth, worldwide passenger traffic will average 5.1 percent growth and cargo traffic will average 5.6 percent growth over the forecast period. To meet this increased demand for air transportation, the number of airplanes in the worldwide fleet will grow at an annual rate of 3.6 percent, nearly doubling from around 19,400 airplanes today to more than 39,500 airplanes in 2030 [1]. This surge in global air traffic coupled with more airplanes going into operation is bound to increase the demand for oil and hence increased costs.

The other issue associated with increased consumption of oil, an issue which has over time become more serious and a major cause of worldwide concern, is that of global warming. With the global surface temperatures on the rise, due to increased green-house gas emissions resulting from the combustion of fossil fuels, nature and the environment has greatly been affected. This issue, if ignored, can lead to grave consequences.

If civil aviation is to continue growing, or even maintain its present level, the industry must address the challenges posed by global warming and increased oil

prices. One solution is provided by unconventional aircraft configurations that significantly improve fuel efficiency and reduce greenhouse gas emissions.

However, modern aircraft are already remarkably efficient. A person traveling from Seattle to New York on a Boeing 737-900 aircraft uses less fuel than one person traveling the same distance in a gas-electric hybrid car. Indeed, the modern aircraft is so highly optimized, that the desired performance improvements are likely not possible using the conventional configuration. This has motivated the investigation of unconventional aircraft and operations [2].

One of the many proposed configuration changes is the concept of active flow control technology. The idea of active control is to expend a small amount of energy to achieve a large gain in performance, such that the total amount of energy used is lower.

1.1 Motivation

Biomimicry or biomimetics is the examination of nature, its models, systems, processes, and elements to emulate or take inspiration from in order to solve human problems. And this has given rise to *biologically inspired engineering*, a new scientific discipline that applies biological principles to develop new engineering solutions.

Right from the time man first tried to attain flight, all the way to the Airbus A380; inspiration has always been drawn from the flight of birds, as they have been considered to be a benchmark for aerodynamically efficient designs. The wing design of the A380 for instance, was inspired by the way the eagle flies. The eagle's wings perfectly balance maximum lift with minimum length. It can manipulate the feathers at its wingtips, curling them to create a 'winglet,' a natural adaptation that aids highly efficient flight. The wingspan of the A380 had to be less than 80m in order to be able to operate on existing international airports. This constraint gave rise to a very important and a challenging question –*How could it generate enough lift and still fit inside airports?* On a conventional wing, vortices created by high pressure air leaks underneath mean the tips do not provide any lift. The wing had to be longer. To tackle this problem, engineers designed small structural devices known as winglets, located on the wing tips, which mimic the eagle's feathers. By doing so, they were able to limit the A380's wingspan to just 79.8 m while generating the desired lift[3].

A swift (*Apus apus* L.) adapts the shape of its wings to the immediate task at hand— folding them back to chase insects, or stretching them out to sleep in flight. During flight, they continually change the shape of their wings from spread wide to swept back. When they fly slowly and straight on, extended wings carry swifts 1.5 times farther and keep them airborne twice as long. Some ten Dutch and Swedish scientists, based in Wageningen, Groningen, Delft, Leiden, and Lund, have shown how wing morphing makes swifts such versatile flyers. Their study proves that swifts can improve flight performance by up to three-fold, numbers that make the wing morphing concept a potential candidate for next generation aircraft designs [4].

1.2 Historical perspective

Even before the official beginning of controlled human flight in 1903, radical shape changing aircraft appeared and then disappeared, contributing little to aviation. Clement Ader, a French inventor and engineer remembered primarily for his pioneering work in aviation, conducted flight experiments in France as early as 1873 and proposed a wing morphing design as early as 1890. He developed ideas for the future of aviation and described them in a short monograph published in 1909. In one of his descriptions of the general military airplane he states -

“Whatever category airplanes might belong to, they must satisfy the following general conditions: their wings must be articulated in all their parts and must be able to fold up completely. When advances in aircraft design and construction permit, the frames will fold and the membranes will be elastic in order to diminish or increase the bearing surfaces at the wish of the pilot...”

In the beginning of the twentieth century many variable geometry aircraft concepts were developed such as Ivan Makhonine’s telescopic wing concept in the 1930s that could change its wingspan by almost 62%. His concept later on inspired Georges Bruner and Charles Gourdou to design a small aircraft named the G-11 C-1. Apparently this airplane was never built, but its design had a wing whose area could range between 11.4 and 17.2 square meters with spans between 6.76 and 11.4 meters. The IS-1 fighter, designed by Nikitin-Shevchenko in 1932, morphed out-of-plane from a bi-plane to a monoplane that was to operate at high speed. The XB-70 supersonic bomber also used a form of three-dimensional wing morphing. This design used outer wing panel rotation to control L/D at both low subsonic and supersonic speeds.

The Mission Adaptive Wing (MAW) project produced wing camber changing concepts and a flight demonstrator, the AFTI/F-111. Variable camber has been shown to improve performance of fighter aircraft at all flight conditions. The F-16 and F-18 aircraft use discrete leading edge and trailing edge flap deflections to control camber, although imperfectly. The MAW supercritical airfoil camber was controlled to create optimum camber over a range of airspeeds for low subsonic to supersonic. This control was achieved by continuously deformable leading and trailing edge deflection using internal mechanisms to bending the chordwise shape as required. This allows the wing to be a continuous surface at all times, unbroken by discrete surfaces[5] .



Figure 1 AFTI-F-111 Mission Adaptive Wing (MAW) in Flight

Lewis et al. patented a variable camber wing command system for varying wing camber to optimize the wing lift-drag ratio during operation of the aircraft.

The invention included a means for sensing various flight conditions and parameters during aircraft operation. The calculating means then calculates a desired position for camber control surface to optimize the wing lift-drag ratio [6].

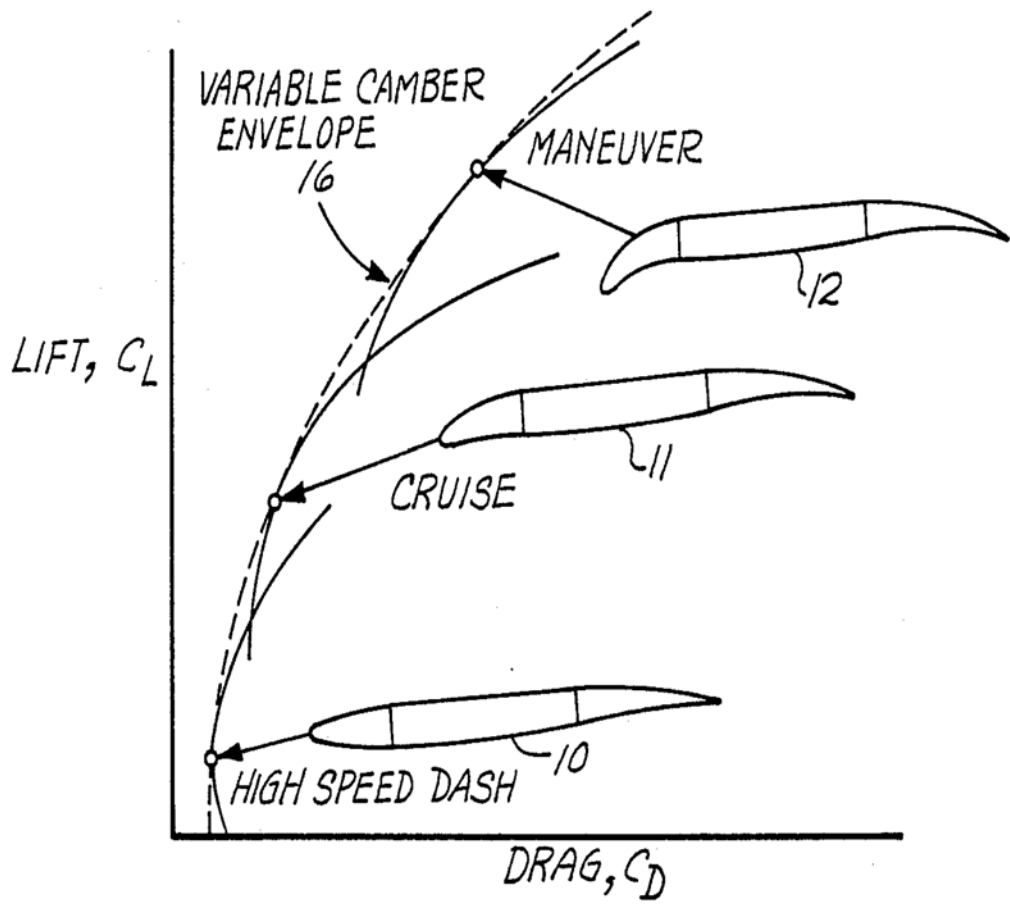


Figure 2 Optimum wing cambers at different flight conditions

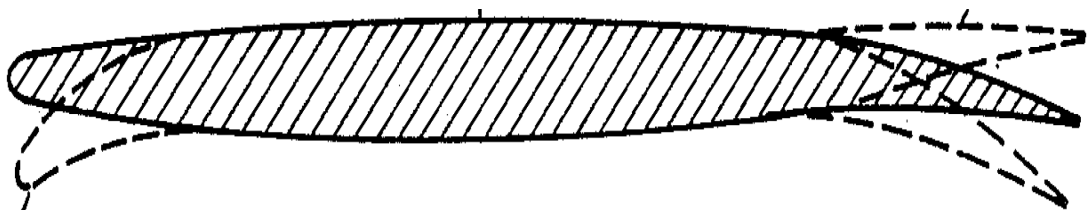


Figure 3 Variable leading and trailing edge wing design

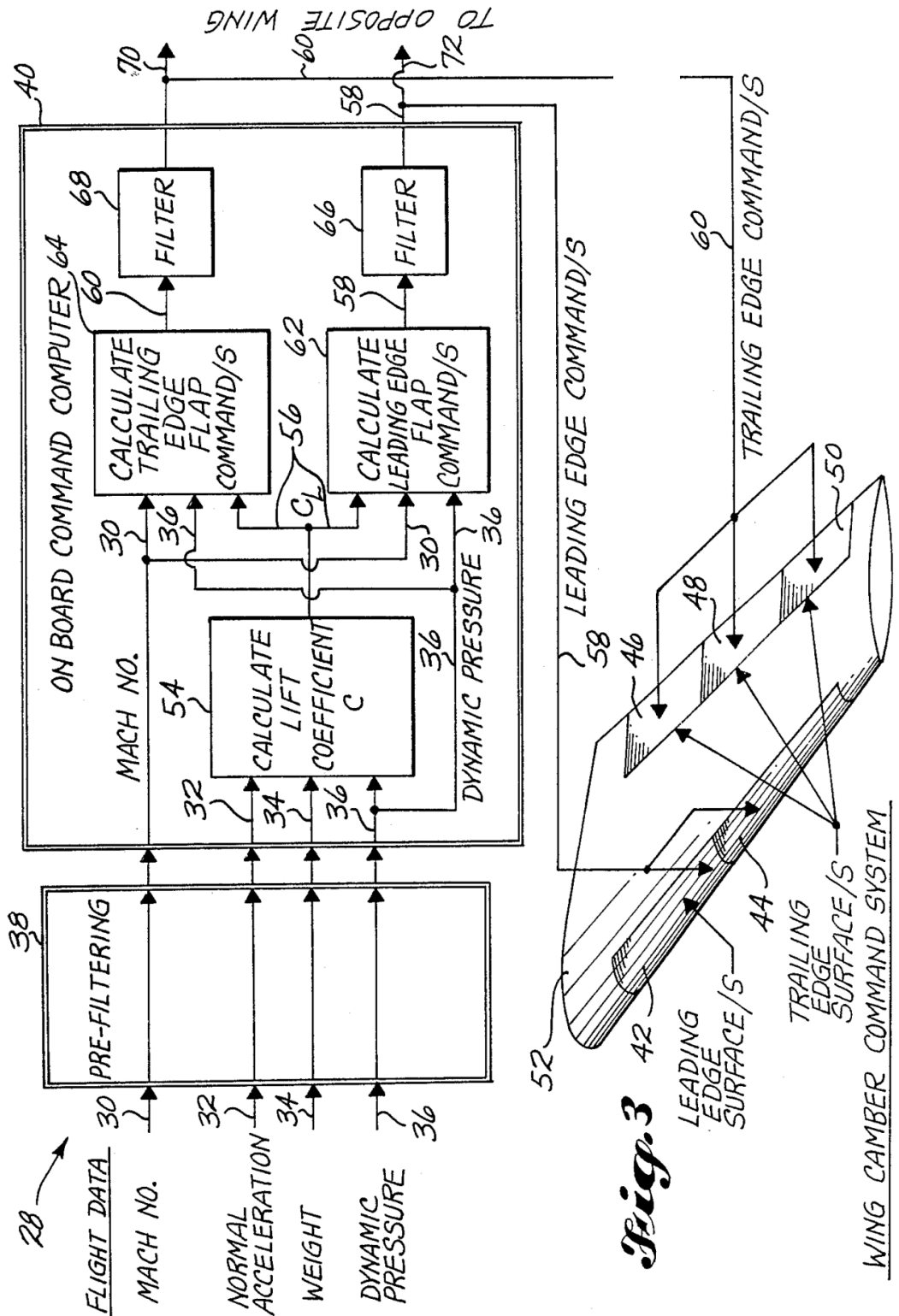


Figure 4 Lewis et al variable camber wing command system

The NASA Aircraft Morphing program is an attempt to couple research across a wide range of disciplines to integrate smart technologies into high payoff aircraft applications. The program bridges research in seven individual disciplines and

combines the effort into activities in three primary program thrusts. System studies are used to assess the highest payoff program objectives, and specific research activities are defined to address the technologies required for development of smart aircraft systems. The stated goal of the program is the development of smart devices using active component technologies to enable self-adaptive flight for a revolutionary improvement in aircraft efficiency and affordability. Aircraft Morphing is an inherently multidisciplinary program, and has been built around a core discipline based structure to provide the fundamental technology base. The integrated disciplines are – Structures; Flow Physics; Systems and Multidisciplinary Optimization; Integration; Controls; Acoustics; and Materials [7] .

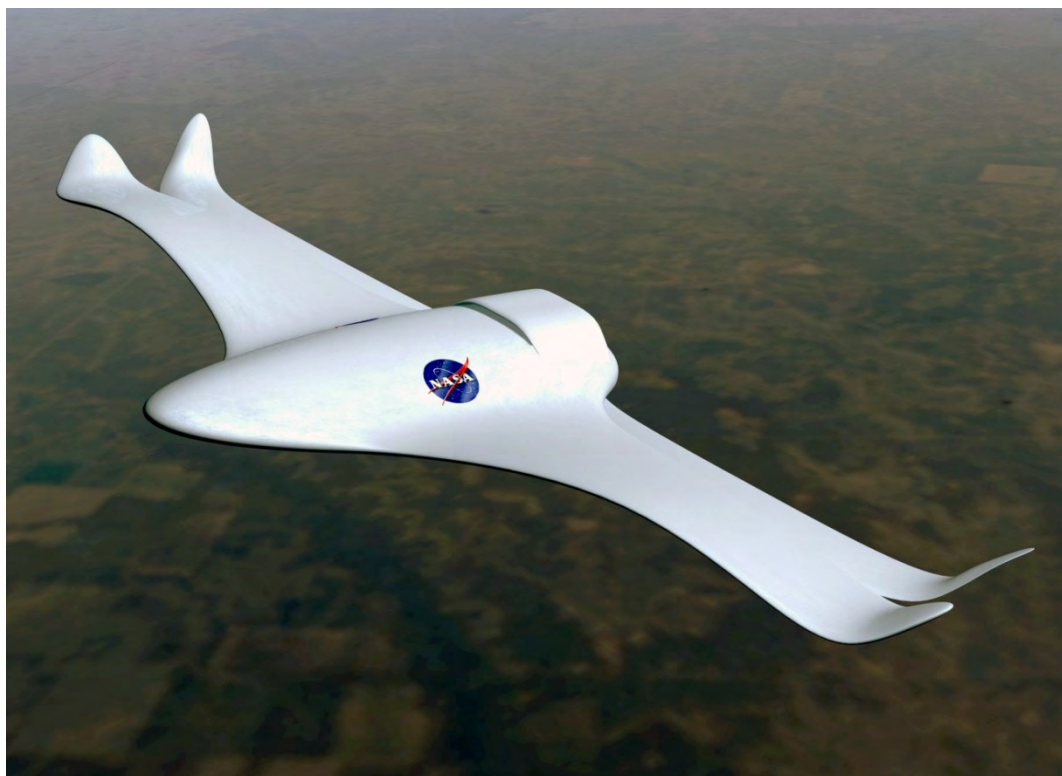


Figure 5 Artist's rendering of NASA's Morphing Airplane

The Boeing Company has also been researching in the field of morphing wings and has published a number of patents pertaining to this subject. Dockter et al. have invented a morphing wing design that includes a rigid internal core, an expandable spar surrounding it and a plurality of elastomeric bladders. An external fiber mesh overlay covers the elastomeric bladders to provide a smooth wing surface. The plurality of elastomeric bladders is expandable through the introduction of

increased air pressure such that the profile of the geometric morphing wing can be modified [8].

In another patent, Sankrithi et al. have invented systems and methods for providing variable geometry winglets to an aircraft. In one of the embodiments, a method for adapting an aircraft to a plurality of flight conditions comprises providing a winglet to each wing. The winglet includes a body portion which in turn includes at least one of a deflectable control surface, a shape memory alloy (SMA) bending plate, and a SMA torque tube. Once the aircraft is moving through the atmosphere, the method further includes providing a winglet modification signal. The winglet is then modified based on the modification signal to at least one of reduce induced drag and redistribute a wing load [9].

The Decadal Survey of Civil Aeronautics conducted by the National Research Council (NRC), has ranked the use of adaptive materials and morphing structures as one of high-priority research and technology (R&T) challenges. This subject has received equal priorities by NASA and at the National level. It has been featured second in the list of R&T challenges in the area of Materials and Structures. DARPA's (Defense Advanced Research Projects Agency) Morphing Aircraft Structures Program has sponsored several wind tunnel experiments at NASA Langley Research Center's Transonic Dynamics Tunnel. These produced a wide range of experience and identified innovative aircraft and rotorcraft concepts and critical materials technologies for future work[10].

1.3 Problem Statement

There are essentially two factors limiting overall aircraft or aircraft-component performance—the development of the boundary layer and the interaction of the boundary layer with the outer flow field, exacerbated at high speeds by the occurrence of shock waves, and the fact that flight and freestream conditions may change considerably during an aircraft mission while the actual aircraft is only designed for multiple but fixed design points rendering such an aircraft a compromise with regard to performance. Flow control and adaptive wings, which are naturally complementary, adjust the flow development to the prevailing flight condition and have, therefore, the potential to greatly improve aircraft and aircraft-component performance[11].

Commercial airplanes generally follow specific flight profiles consisting of take-off, climb, cruise, descend and landing. These flight profiles essentially change the freestream conditions in which the aircrafts operate. Furthermore, over the course of the flight, the required lift force changes as the fuel gets consumed. The conventional fixed wing designs account for these requirements by catering to multiple but fixed design points which, however, compromise the overall flight performance.

The two dominant flight parameters influencing aerodynamic forces of a given wing are—airspeed and altitude. It is desirable that the aerodynamic efficiency of the wing be at its maximum (subject to certain constraints) throughout the flight envelope. Employing adaptive wing technology where the effective wing geometry adjusts to the changing flow and load requirements, allows to fully exploring the aerodynamic flow potential at each point of the flight envelope, thus resulting in aerodynamic performance gains during cruise and maneuver, and, furthermore, most likely improving structural designs.

A solution to this problem is the morphing wing concept in which the wing shape is altered depending on the existing flight condition. The primary motive for altering wing geometry is to improve airfoil efficiency in off-design flight regimes. A particular, more constrained implementation of this concept can be seen in most modern aircraft designs existing today and it takes the form of flaps which change the wing area and/or effective camber. A polymorph wing (variable planform) and variable pitch or incidence are also proven methods of wing adaptation and can be found in some multi-role combat aircrafts [12]. However these forms essentially allow only rigid or constrained deformations in that, the deformations are discrete or discontinuous such as the case of trailing edge flaps or leading edge slats deployed while take-off and landing in transport aircrafts. In the remaining portion of the flight the wing section is predominantly constant whereas the free-stream conditions continue to change with change in airspeed and altitude thus compromising all round performance. In order to improve the performance over a greater portion of the flight envelope, it is desirable that the wing shape undergoes considerable deformation thereby morph into a shape that achieves desirable low-drag pressure distributions.

The problem definition for this thesis is therefore to develop an optimization framework that could be used to develop single-point optimized airfoil shapes at

discrete locations within the flight envelope. These shapes could then be embedded in a control system architecture that could signal actuators to deform the existing airfoil shape into the desired one that has been optimized for the corresponding freestream conditions of the aircraft.

1.4 Review of Literature

A plethora of research and studies has been performed in the area of aerodynamic shape optimization for the design of optimal airfoil shapes. This research area has undoubtedly been dominated by the pioneering works of Antony Jameson who has extensively used control theory for airfoil and wing design [13-16]. Here the shape optimization problem is treated as a control problem, in which the airfoil (and/or wing) is treated as a device which controls the flow to produce lift with minimum drag, while meeting other requirements such as low structure weight, sufficient fuel volume, and stability and control constraints. The theory of optimal control of systems is applied which is governed by partial differential equations with boundary control, in this case through changing the shape of the boundary. Here the Frechet derivative of the cost function is determined via the solution of an adjoint partial differential equation, and the boundary shape is then modified in a direction of descent. This process is repeated until an optimum solution is approached.

E. Stanewsky [11] provides a thorough review of the adaptive wing technology. Here, various concepts on complete wing adaptation both for cruise-drag minimization as well as for maneuver load control for combat aircrafts are discussed. The concepts that have been discussed are the adaptive variable camber trailing edges, leading edges and transonic contour bumps, and also combinations of the three. He also discusses optimum number of linear actuators based on the drag reduction achieved and the percentage flexing of the surfaces.

Fred Austin et al. have developed optimum wing cross-sectional profiles for a hypothetical fixed-wing version of the F-14 aircraft at two transonic cruise conditions [17]. Jameson's control-theory based optimum aerodynamic design approach is used for the analysis at critical operating conditions where shape modifications might significantly improve the aerodynamic performance. Based on the analysis it is shown that only small, potentially achievable, adaptive modifications to the profile are required. A general method is developed to adaptively deform the structures to

desired shapes. The method is investigated by a finite element analysis of an adaptive rib and by experimental investigations on a rib model. Open loop experiments of the unloaded structure show that commanded shapes can be achieved.

David Zingg et al. have used a Newton-Krylov based aerodynamic optimization algorithm to assess an adaptive airfoil concept for drag reduction at transonic speeds [18]. The lift-constrained drag minimization problem is used as the objective function. The airfoil geometry is parameterized using B-splines and the compressible Navier-Stokes equations are solved with a Newton-Krylov method. In this study, a baseline is designed to produce low drag at a fixed lift coefficient over a range of Mach numbers. This airfoil is then compared with a sequence of nine airfoils, each designed to be optimal at a single operating point in the Mach number range. It is found that shape changes of less than 2% chord lead to drag reduction of 4 to 6% over a range of Mach numbers from 0.68 to 0.76. If the shape changes are restricted to the upper surface only, then changes of less than 1% chord lead to a drag reduction of 3 to 5%.

Namgoong in his research has developed appropriate problem formulations and optimization strategies to design an airfoil for morphing aircraft that include the energy required for shape change [19]. Here the relative strain energy needed to change from one airfoil shape to another is included as an additional design objective along with a drag design objective, while constraints are enforced on the lift. Solving the resulting multi-objective problem generates a range of morphing airfoil designs that represent the best tradeoffs between aerodynamic performance and morphing energy requirements. The airfoil parameterization is done here using Hicks-Henne shape functions. To formulate the problem as a multi-objective problem, a strain energy model is developed that depends upon the concept of linearly elastic springs. The Genetic Algorithm (GA) is used for solving the multi-objective optimization problem.

Gamboa et al. in their work have designed a morphing wing concept for a small experimental unmanned aerial vehicle to improve the vehicle's performance over its intended speed range [20]. The wing is designed with a multidisciplinary design optimization tool, in which an aerodynamic shape optimization code coupled with a structural morphing model is used to obtain a set of optimal wing shapes for minimum drag at different flight speeds. The optimization studies reveal that wing

drag reduction of up to 30% can be achieved by morphing the wing, thus representing the important performance improvements in off-design conditions.

Ursache in his research has designed adaptive structures for achieving global multi-shape morphing aerodynamic configurations, by using slender structures. A heuristic approach is proposed that enables morphing through a range of stable cambered airfoils to achieve the aerodynamic properties for different maneuvers, with the benefit of low powered actuation control [21]. Ünlüsoy in his work has performed structural design and analysis of a wing having mission adaptive control surfaces. The wing is designed to withstand a maximum aerodynamic loading of 5g during maneuver. The structural model of the wing is developed using MSC/PATRAN and the analysis is carried out using MSC/NASTRAN [22].

1.5 Thesis Objectives

As outlined in the previous section, the goal is to achieve a set of airfoil shapes that have been optimized for specific points in the flight envelope, viz at specific airspeeds and altitudes.

This, as is evident from the aforementioned description, essentially translates into an optimization problem. Thus, aerodynamic shape optimization forms the core and one of the most important parts of this thesis which, in the subsequent chapters, will be addressed in greater detail.

The shape optimization process is to be carried out for different operating points within the flight envelope. Only those operating points are chosen which have a greater probability of being visited by a typical subsonic transport airplane. A comparative analysis is performed in which the aerodynamic performance of a Boeing-737 classic airfoil at various operating points is compared to that of a single point optimized airfoil at the same operating points.

Chapter 2 provides an overview of the theories required for the shape optimization process. This includes the formulation of the optimization problem, shape parameterization to obtain the design variables, various objective functions that are employed for shape optimization problems and the constraints on the optimization process. The governing equations that form an important part of any fluid dynamics

problem alongwith the procedure for analysis using Computation Fluid Dynamics (CFD) solvers are also discussed.

Chapter 3 introduces the Iterative Response Surface Optimization methodology for carrying out the shape optimization process. Here the various steps involved such as the Design of Experiments (DoE), response surface construction and the iterative improvement of this model are discussed. Also included is the model testing methods and the mathematical optimization of the response model. This is followed by the steps for constructing the CFD grid using GridPro® and the flow analysis using ANSYS-Fluent® along with its validation are discussed towards the end of the chapter.

Chapter 4 primarily discusses the results of this study. Here the lift-moment constrained drag minimization scheme employed in the optimization process is discussed followed by the case study. Also discussed are the operational flight envelopes of typical transonic aircrafts. The RAE 2822 design study is carried out to validate the optimization algorithm developed. The adaptive airfoil concept is demonstrated using a Boeing-737 classic airfoil at three steady flight operating conditions. The aerodynamic performance of the adaptive airfoil is then compared to that of the baseline airfoil.

Chapter 5 summarizes and concludes the research conducted in this thesis along with shedding some light on possible improvements and more advanced study that can be carried out on this subject in the future.

2. OPTIMIZATION AND CFD

2.1 Introduction to Optimization

Nature optimizes. Physical systems tend to a state of minimum energy. The molecules in an isolated chemical system react with each other until the total potential energy of their electrons is minimized. Rays of light follow paths that minimize their travel time.

On similar lines people optimize. Manufacturers aim for maximum efficiency in the design and operation of their production processes. Engineers adjust parameters to optimize the performance of their designs [23].

Optimization is the act of obtaining the best result under given circumstances. In design, construction, and maintenance of any engineering system, engineers have to take many technological and managerial decisions at several stages. The ultimate goal of all such decisions is either to minimize the effort required or to maximize the desired benefit [24]. Optimization can be considered as an important tool in decision science and in the analysis of physical systems. To make use of this tool, an *objective* must be defined. The objective is a quantitative measure of the performance of the system under study. This could be profit, time, potential energy, or any quantity or combination of quantities that can be represented by a single number. The objective depends on certain characteristics of the system, called *variables* or *unknowns*. The goal is to find values of the variables that optimize the objective. Often the variables are restricted, or *constrained*, in some way.

The process of identifying objective, variables, and constraints for a given problem is known as *modeling*. Construction of an appropriate model is the first and one of the most important step in the optimization process. Once the model has been formulated, an optimization algorithm can be used to find its solution, usually with the help of a computer. There is no universal optimization algorithm but rather a collection of algorithms, each of which is tailored to a particular type of optimization problem. The responsibility of choosing the algorithm that is appropriate for a specific application often falls on the user. This choice is an important one, as it may determine whether the problem is solved rapidly or slowly and, indeed, whether the solution is found at all.

2.1.1 Mathematical Formulation

Mathematically, optimization is the minimization or maximization of a function subject to constraints on its variables. The optimization problem can be written as follows:

$$\min_{\mathbf{X} \in \mathbb{R}^n} f(\mathbf{X}) \quad \text{subject to} \quad \begin{aligned} g_i(\mathbf{X}) &= 0, \quad i = 1, 2, \dots, m \\ h_k(\mathbf{X}) &\leq 0, \quad k = 1, 2, \dots, p \end{aligned}$$

- \mathbf{X} is the vector of variables, also called unknowns or parameters;
- f is the objective function, a scalar function of \mathbf{X} that has to be optimized.
- g_i and h_k are the constraint functions, which are scalar functions of \mathbf{X} that define certain equalities and inequalities that \mathbf{X} must satisfy.
- m and p are the number of equalities and inequalities constraints.

2.1.2 Characteristics of Optimization Algorithms

Optimization algorithms are iterative. They begin with an initial guess of the variable \mathbf{X} and generate a sequence of improved estimates, called *iterates*, until they terminate, hopefully at a solution. The strategy used to move from one iterate to the next distinguishes one algorithm from another. Most strategies make use of the values of the objective function f , the constraint functions g_i and/or h_k , and possibly the first and second derivatives of these functions. Some algorithms accumulate information gathered at previous iterations, while others use only local information obtained at the current point.

Regardless of these specifics, good algorithms should possess the following properties:

- *Robustness*: They should perform well on a wide variety of problems in their class, for all reasonable values of the starting point.
- *Efficiency*: They should not require excessive computer time or storage.
- *Accuracy*: They should be able to identify a solution with precision, without being overly sensitive to errors in the data or to the arithmetic rounding errors that occur when the algorithm is implemented on a computer.

These goals may conflict. For example, a rapidly convergent method for a large unconstrained nonlinear problem may require too much computer storage. On

the other hand, a robust method may also be the slowest. Tradeoffs between convergence rate and storage requirements, and between robustness and speed, and so on, are central issues in numerical optimization.

2.2 Sequential Quadratic Programming

Sequential quadratic programming (SQP) is an iterative method for nonlinear optimization. SQP methods are used on problems for which the objective function and the constraints are twice continuously differentiable.

The sequential quadratic programming is one of the most recently developed and perhaps one of the best methods of optimization. The method has a theoretical basis that is related to (1) the solution of a set of nonlinear equations using Newton's method, and (2) the derivation of simultaneous nonlinear equations using Kuhn–Tucker conditions to the Lagrangian of the constrained optimization problem.

2.2.1 The SQP Technique

Consider a nonlinear optimization problem as below:

$$\min_{\mathbf{X}} f(\mathbf{X})$$

Subject to $h_k(\mathbf{X}) \leq 0, k = 1, 2, \dots, p$

The Lagrange formulation of the above problem then becomes

$$L(\mathbf{X}, \lambda) = f(\mathbf{X}) + \sum_{k=1}^p \lambda_k h_k(\mathbf{X})$$

Here λ_k are the Lagrange multipliers for each of the k th constraints. It is assumed that the bound constraints have been expressed as inequality constraints. Now in the SQP technique, the above optimization problem can be solved iteratively by solving a Quadratic Programming (QP) problem which is stated as—

$$\min_{\Delta \mathbf{X}} Q = \nabla f^T \Delta \mathbf{X} + \frac{1}{2} \Delta \mathbf{X}^T [\nabla^2 L] \Delta \mathbf{X}$$

subject to

$$g_j + \nabla g_j^T \Delta \mathbf{X} \leq 0, \quad j = 1, 2, \dots, m$$

$$h_k + \nabla h_k^T \Delta \mathbf{X} = 0, \quad k = 1, 2, \dots, p$$

The Lagrange formulation is given by

$$\tilde{L} = f(\mathbf{X}) + \sum_{j=1}^m \lambda_j g_j(\mathbf{X}) + \sum_{k=1}^p \lambda_{m+k} h_k(\mathbf{X})$$

Since the minimum of the augmented Lagrange function is involved, the SQP method is also known as the projected Lagrangian method.

2.2.2 Solution Procedure

As in the case of Newton's method of unconstrained minimization, the solution vector $\Delta\mathbf{X}$ is treated as the search direction, \mathbf{S} , and the quadratic programming sub-problem (in terms of the design vector \mathbf{S}) is restated as:

$$\min_{\mathbf{S}} Q(\mathbf{S}) = \nabla f(\mathbf{X})^T \mathbf{S} + \frac{1}{2} \mathbf{S}^T [H] \mathbf{S}$$

Subject to

$$g_j(\mathbf{X}) + \nabla g_j(\mathbf{X})^T \mathbf{S} \leq 0, \quad j = 1, 2, \dots, m$$

$$h_k(\mathbf{X}) + \nabla h_k(\mathbf{X})^T \mathbf{S} = 0, \quad k = 1, 2, \dots, p$$

where $[H]$ is a positive definite matrix that is taken initially as the identity matrix and is updated in subsequent iterations so as to converge to the Hessian matrix of the Lagrange function \tilde{L} . The above sub-problem is a quadratic programming problem and can be solved using any optimization technique. Once the search direction \mathbf{S} , is found the design variables can be updated as

$$\mathbf{X}_{j+1} = \mathbf{X}_j + \alpha \cdot \mathbf{S}$$

Here α is the optimal step length along the direction \mathbf{S} and can be determined by an appropriate line search procedure so that a sufficient decrease in a merit function is obtained. The matrix $[H]$ is a positive definite approximation of the Hessian matrix of the Lagrangian function.

2.2.3 Updating the Hessian Matrix

At each major iteration a positive definite quasi-Newton approximation of the Hessian of the Lagrangian function, $[H]$, is calculated using the BFGS (Broyden-Fletcher-Goldfarb-Shanno) method, where $\lambda_i, i = 1, \dots, m$, is an estimate of the Lagrange multipliers.

$$H_{k+1} = H_k + \frac{q_k q_k^T}{q_k^T s_k} - \frac{H_k^T s_k^T s_k H_k}{s_k^T H_k s_k},$$

Where

$$s_k = \mathbf{X}_{k+1} - \mathbf{X}_k$$

$$q_k = \left(\nabla f(\mathbf{X}_{k+1}) + \sum_{i=1}^m \lambda_i g_i(\mathbf{X}_{k+1}) \right) - \left(\nabla f(\mathbf{X}_k) + \sum_{i=1}^m \lambda_i g_i(\mathbf{X}_k) \right)$$

2.3 Design Variables

As mentioned in the previous section, design variables are quantities or quantifiable characteristics on which the performance of the system depends. These are quantities which almost completely define the system being analyzed.

From an aerodynamic shape optimization point of view, the vector of design variables, \mathbf{X} , primarily contains parameters that control the shape of the airfoil. These are geometrical entities that are relatively few in number and fully parameterize the airfoil shape. The angle of attack, which is the angle that the airfoil chordline makes with the direction of airflow, may be used as an additional design variable.

Here, B-Spline control points are used to define the airfoil shape. These control points along with angle of attack form the vector of design variables for the purpose of optimization. The following sub-sections describe the process of shape parameterization using B-Spline curves.

2.3.1 B-Spline Curves Introduction

NURBS, or Non-Uniform Rational B-Splines, are the standard for describing and modeling curves and surfaces in computer aided design and computer graphics. They are used to model everything from automobile bodies and ship hulls to animated characters.

B-Splines belong to the class of parametric curves. Parametric curve representations of the form: $x = f(t)$; $y = g(t)$; $z = g(t)$

where t is the parameter, are extremely flexible. They are axis independent, easily represent multiple-valued functions and infinite derivatives, and have additional degrees of freedom compared to either explicit or implicit formulations.

2.3.2 B-Spline Formulation

The general mathematical formulation of a B-spline curve given the order, control point coordinates and a parameter set, is as follows:

$$C_x(t_j) = \sum_{i=1}^{n+1} N_{i,k}(t_j) \cdot P_i^x$$

$$C_y(t_j) = \sum_{i=1}^{n+1} N_{i,k}(t_j) \cdot P_i^y$$

The notations used in the above formulation are as described below:

- (C_x, C_y) are the Cartesian coordinates of the B-Spline curve.
- t is the curve parameter,
- (P_i^x, P_i^y) are the co-ordinates of the B-Spline control points,
- $(n+1)$ is the number of control points,
- k is the order of the curve (degree is $k-1$),
- j is the index for the curve parameter u ,
- $N_{i,k}$ are the B-Spline Basis functions defined by the *Cox-de Boor* recurrence relations as below:

$$N_{i,1}(t_j) = \begin{cases} 1, & \text{if } U_i \leq t_j < U_{i+1} \\ 0, & \text{otherwise} \end{cases}$$

$$N_{i,k}(t_j) = \frac{t_j - U_i}{U_{i+k-1} - U_i} N_{i,k-1}(t_j) + \frac{U_{i+k} - t_j}{U_{i+k} - U_{i+1}} N_{i+1,k-1}(t_j) \quad]$$

- The vector U_i represents a normalized uniform knot sequence given by:

$$U_i = \begin{cases} 0, & \text{for } 1 < i \leq k \\ (U_{i-1} + 1)/(U_{(n+1)+k}), & \text{for } k < i < (n+1) + 2 \\ U_{i-1}/(U_{(n+1)+k}), & \text{for } (n+1) + 2 \leq i \leq (n+1) + k \end{cases}$$

2.3.3 Airfoil parameterization using B-Spline curves

The distance along the B-spline curve is represented by the parameter value t_j for each point j on the surface of the airfoil. The initial values of t_j are given by

$$t_1 = 0$$

$$t_j = \frac{\sum_{i=2}^j \sqrt{(x_{data_i} - x_{data_{i-1}})^2 + (y_{data_i} - y_{data_{i-1}})^2}}{L_T} \dots j \in [2, N_{data}]$$

$$L_T = \sum_{i=2}^{N_{data}} \sqrt{(x_{data_i} - x_{data_{i-1}})^2 + (y_{data_i} - y_{data_{i-1}})^2} \dots (x_{data_i}, y_{data_i})$$

$$\in [0,1]$$

Here:

- N_{data} , is the number of data points of the airfoil to be parameterized,
- (x_{data}, y_{data}) are the airfoil coordinates.
- L_T is the sum of the chord distances for all the data points.

At the onset of the optimization procedure, it is necessary to determine the location of the B-spline control points that best approximate the initial airfoil shape. In order to locate the control point co-ordinates a curve-fitting strategy must be employed. The curve fitting approach employed here implements a curve parameter correction algorithm [25]. The algorithm consists of the following two separate parts

- Initial parameterization
- Fitting/reparameterization loop

Initial parameterization yields an initial set of control-point coordinates that roughly approximates the airfoil shape. If a data point lies on the B-spline curve, then it must satisfy—

$$\begin{aligned} C_x(t_1) &= N_{1,k}(t_1) \cdot P_1^x + N_{2,k}(t_1) \cdot P_2^x + \dots + N_{n+1,k}(t_1) \cdot P_{n+1}^x C_y(t_1) \\ &= N_{1,k}(t_1) \cdot P_1^y + N_{2,k}(t_1) \cdot P_2^y + \dots + N_{n+1,k}(t_1) \cdot P_{n+1}^y \\ C_x(t_2) &= N_{1,k}(t_2) \cdot P_1^x + N_{2,k}(t_2) \cdot P_2^x + \dots + N_{n+1,k}(t_2) \cdot P_{n+1}^x C_y(t_2) \\ &= N_{1,k}(t_2) \cdot P_1^y + N_{2,k}(t_2) \cdot P_2^y + \dots + N_{n+1,k}(t_2) \cdot P_{n+1}^y \vdots \\ C_x(t_j) &= N_{1,k}(t_j) \cdot P_1^x + N_{2,k}(t_j) \cdot P_2^x + \dots + N_{n+1,k}(t_j) \cdot P_{n+1}^x C_y(t_j) \\ &= N_{1,k}(t_j) \cdot P_1^y + N_{2,k}(t_j) \cdot P_2^y + \dots + N_{n+1,k}(t_j) \cdot P_{n+1}^y \end{aligned}$$

where $2 \leq k \leq n + 1 \leq j$. This system of equations is more compactly written in matrix form as

$$\begin{aligned} [C_x] &= [N] \cdot [P_x] \\ [C_y] &= [N] \cdot [P_y] \end{aligned}$$

Recalling that a matrix times its transpose is always square, the control polygon for a B-spline curve that approximates the data is given by

$$[N]^T[C_x] = [N]^T[N] \cdot [P_x]$$

$$\therefore [P_x] = [[N]^T[N]]^{-1} \cdot [C_x]$$

Likewise

$$[N]^T[C_y] = [N]^T[N] \cdot [P_y]$$

$$\therefore [P_y] = [[N]^T[N]]^{-1} \cdot [C_y]$$

Provided that the order of the B-spline basis k , the number of control polygon vertices $n + 1$, and the parameter values along the curve are known, then the basis functions $N_{i,k}(t_j)$ and hence the matrix $[N]$ can be obtained. Within the restrictions $2 \leq k \leq n + 1 \leq j$, the order and number of polygon vertices are arbitrary.

In the *reparameterization* stage, the parameter values are improved by replacing the previous parameters by new parameter values which are obtained after implementing parameter correction formula. The algorithm is explained as follows.

The basic aim here is to fit a B-Spline curve that will approximate N measured data points in a Least Squares Method sense. This leads to minimization problem in which the objective is to find an optimal set of parameter values t_i producing an optimal approximating spline $C(t_i)$ with minimal distances to the data points D_i . The objective function is—

$$E_i = \sum_{i=1}^N \|D_i - C(t_i)\|^2 \rightarrow \min$$

Where $C(t_i)$ is the B_spline curve point at t_i and D_i is the corresponding measured data point. In curve parameterization strategies, the distance vectors are generally not perpendicular to the surface. This means an arbitrary error due to the parameterization has been minimized. This parameterization scheme can be improved by changing the aim to approximate the shortest distances, which means sequence of new parameter values t_i will be constructed with the goal that the corresponding error vectors E_i converge in general to the normal of the approximation curve.

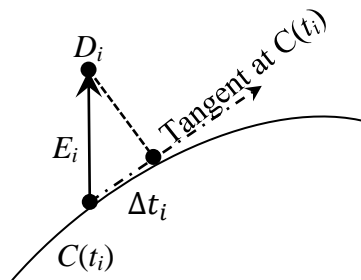


Figure 6 Reparameterization

Therefore, the curve at each point $C(t_i)$ is replaced by the tangent as described in Fig.2.1, by projecting the error vector E_i on the tangent and obtain Δt_i as a measure for changing the parameter values t_i in direction of the parameter values of the perpendicular from D_i to $C(t_i)$. One of the methods for minimizing the local error vector is as suggested by Hoschek et al [26]. The idea is to minimize $E_i^2 = (D_i - C(t_i))^2$. This after differentiation leads to: $f := \langle D_i | \dot{C}(t_i) \rangle = 0$

Now if the Newton iteration formula is used to compute a zero of f , the correction formula is obtained as

$$\Delta u_i = -\frac{f}{\dot{f}} = \frac{-\langle D_i | \dot{C}(t_i) \rangle}{\langle D_i | \ddot{C}(t_i) \rangle - \langle D_i | D_i \rangle} = -\frac{(C(t_i) - D_i) \cdot \dot{C}(t_i)}{(C(t_i) - D_i) \cdot \ddot{C}(t_i) + \dot{C}(t_i)^2}$$

Therefore the improved parameter becomes $t'_i = t_i + \Delta t_i$

This reparameterization step is repeated till the desired accuracy is achieved. The least-squares problem is re-evaluated with the final parameter vector in order to obtain a better set of control points. This procedure typically converges within a few hundred iterations and do not require significant computational effort.

In addition, the airfoil geometry to be parameterized is split into two —*upper surface* and *lower surface*. This way the parameterization is split into two parts—one for each of the split surfaces. So a set of B-Spline control points and a set of curve parameters control one of the two surfaces of the airfoil, and a different pair of sets controls the other airfoil surface. The shape control is hence independent for the upper and lower surfaces, thus offering greater flexibility.

The parameterization is carried out in a way so that the approximating B-Spline curve passes through the first and the last control points, which form the trailing edge and leading edge of the airfoil respectively. This ensures that B-Spline curve always passes through the leading and trailing edges thus maintaining a fixed chord length. Also the B-spline curve maintains tangency to the control points polygon at the first and last control points. This behavior of the B-spline curve is as a result of the formulation of the uniform knot vector which has been described in Section 2.2.2.

Each of the two airfoil surfaces is controlled individually by seven B-spline control points and a set of curve parameters. The numbering of the control points is

done counter-clockwise beginning with the upper surface and then moving on the lower surface. For the upper surface the numbering begins with control point number 1 at the trailing edge ending with the control point number 7 at the leading edge. For the lower surface the counting begins with control point number 8 at the leading edge ending with control point number 14 at the trailing edge. The first and the seventh control points remain fixed for the upper surface as they represent the trailing and leading edges respectively, as described above. Likewise, eighth and the fourteenth control point remain fixed for the lower surface. Control points 1 and 14 (trailing edge); 7 and 8 (leading edge) are coincident. For the upper surface, control point 6 and 7 share the same x-ordinate. Likewise, for the lower surface, control point 8 and 9 share the same x-ordinate. This is done so as to ensure a smooth and continuous transition from the upper surface to the lower surface at the leading edge. By doing so, there are no cusps or kinks formed, as both, the upper and lower surfaces have the same tangency at the leading edge control points. It may be noted here that, as the parameterization is carried out individually for the upper and lower surface of the airfoil, there are two different sets of curve parameters u_i (one for each surface). These two sets remain fixed throughout the optimization process.

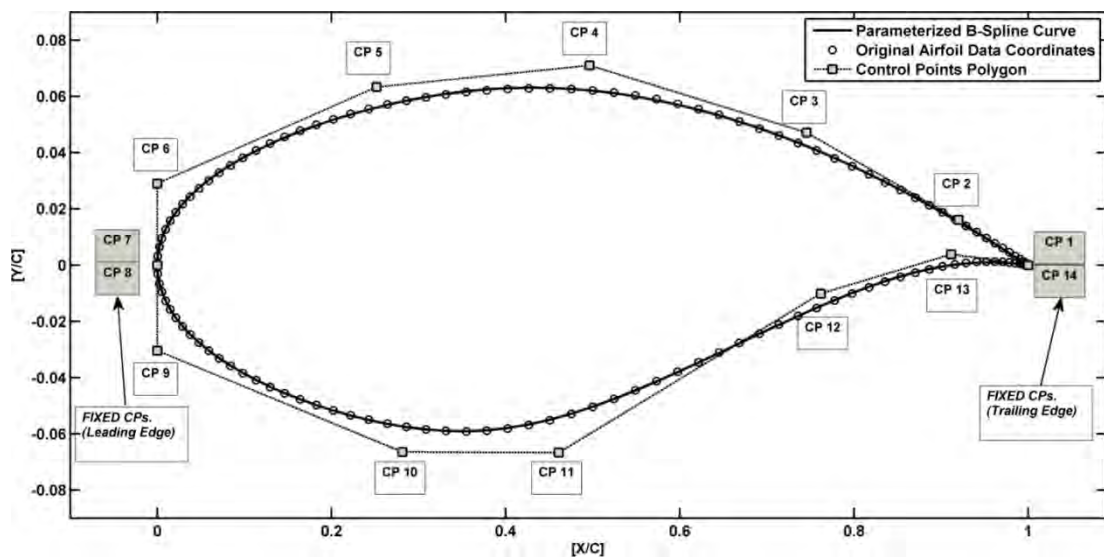


Figure 7 B-Spline parameterization of RAE 2822 airfoil

As an example, the parameterization of RAE 2822 airfoil is shown in the above figure. The original airfoil data is split into upper and lower surfaces. The parameterization is then carried out using 14 control points (7 for each surface) and curve order of 5. CPs 1, 14, 7 and 8 are fixed. CPs 6 and 9 share the same x-ordinates

as those of CPs 7 and 8. The algorithm fits well to the airfoil data as is evident from the above figure.

2.3.4 Design Variables Vector

The design variables vector is formed by concatenating the y-ordinates of all the free controls points— CPs 2 through 6 (for the upper surface) and CPs 9 through 13 (for the lower surface). In addition to the ten control points, the angle of attack is included as the eleventh design variable. As is evident, the CPs are allowed to move only along the y-direction and so the x-ordinates remain fixed throughout the optimization process. However, if required, the x-ordinates can readily be added to the design variables vector.

2.4 Objective Functions

Objective function as is described earlier is a scalar valued function that determines the performance of the system. From an aerodynamic shape optimization point of view, the objective is primarily to minimize the drag or maximize the aerodynamic efficiency.

There are various objective functions that cater to the problem of aerodynamic shape optimization as discussed below:

2.4.1 Inverse design

In the inverse design formulation, the objective is to reach a shape such that the pressure distribution over its surface matches a target pressure distribution. Mathematically, the objective is represented as

$$J = \frac{1}{2} \int_{T.E}^{L.E} \left(C_P(s_{u,l}) - C_P^*(s_{u,l}) \right)^2 ds_{u,l}$$

where C_P^* represents the target pressure coefficient as function of $s_{u,l}$ is the distance along the upper and lower surfaces measured from the Teading edge (L.E) to the Leading Edge. By minimizing J , the optimizer finds the shape of the airfoil that, in the least-squares sense, best matches the target pressure distribution. A drawback associated with this type of formulation is that the target pressure distribution needs to be provided by the user. This requires considerable amount of experience and knowledge of the flow physics on the part of the user.

2.4.2 Maximizing Aerodynamic Efficiency

This is a relatively easier function, where the objective is to simply maximize the aerodynamic efficiency which is the ratio of the lift to the drag. The problem is generally formulated as the minimization of the inverse of the aerodynamic efficiency.

$$J = \frac{C_D}{C_L}$$

This objective is suitable for problems which do not require a constraint on the lift force or moments.

2.4.3 Lift-Constrained Drag Minimization

In the lift-constrained drag-minimization formulation, the objective is to reduce drag while maintaining the desired lift force. This type of formulation is suitable for problems involving shape optimizations for airplanes where it is advantageous to reduce the drag while ensuring that the lift generated is greater than and approximately equal to the weight of the airplane in steady flight. Mathematically, the problem is formulated as

$$J = \begin{cases} \omega_L \left(1 - \frac{C_L}{C_L^*}\right)^2 + \omega_D \left(1 - \frac{C_D}{C_D^*}\right)^2 & \text{if } C_D > C_D^* \\ \omega_L \left(1 - \frac{C_L}{C_L^*}\right)^2 & \text{otherwise} \end{cases}$$

where C_L^* and C_D^* represent the target lift and drag coefficients, respectively. The coefficients ω_L and ω_D are the weights on the lift and drag coefficients respectively. C_D^* is often selected low enough to be considered unattainable. This objective function consists of two competing objectives, namely attaining the target lift and minimizing drag. Achieving both simultaneously is difficult, since a decrease in drag typically corresponds to a decrease in lift. As the lift is a constraint on the design, it is important to ensure that the desired coefficient of lift is reached. To do this, either the weight on the lift coefficient objective must be very high (which will reduce the amount by which drag is decreased), or the target lift entered into the optimization program must be a carefully selected amount greater than the actual target lift [27].

To determine how much greater the target lift coefficient given to the program should be than the actual desired lift coefficient requires some experience. Some level of experience is also required to successfully select a value for C_D^* , and selecting appropriate parameters can take several trials.

One of the advantages of using this kind of formulation is that additional terms can be readily added to the problem. For example the moment coefficient along with a suitable weight, can be added in the same manner as the lift coefficient. Also this formulation falls into the category of unconstrained minimization in which the constraint is included within objective function as a penalty term. Here the constraint on the lift coefficient is added as penalty term in the objective function. Doing so avoids an extra formulation of a constraints equation thereby reducing the burden on the optimization process.

In this thesis, the lift- moment constrained drag-minimization objective function will be employed.

2.5 Flow Equations

In order to obtain values of lift, drag and moment coefficients as well as the surface pressure distribution, which are needed to evaluate the objective functions, it is necessary to compute the flow or in other words determine the solution of the airflow over the surfaces.

From fluid mechanics, there are fundamental equations that govern the behavior of any fluid that flows over a submerged surface. These are the governing equations that derived using the three fundamental physical principles that central to the macroscopic observations of nature, which are:

- Conservation of mass, i.e., mass can neither be created nor be destroyed.
- Newton's Second Law, i.e., force is directly proportional to the rate of change of momentum, and
- Conservation of energy, i.e., energy can only change from one form to another.

These principles have led to the formulation of complex differential equations that essentially describe the governing of fluid motion. These equations are:

- Continuity and Momentum Equations which are known as the Mass Conservation Equation and the Momentum Conservation Equation respectively.
- Energy Conservation Equation to model the compressibility effects.

These equations must be solved in order to obtain the solution of the flow which is basically the lift, drag, moment, pressure distribution etc. In addition to these fundamental equations, certain transport equations must be solved to model the turbulence which is an important characteristic associated with high Reynolds number flows. The details and derivations of the fundamental and the additional transport equations can be found in various literatures [28]. Here only the equations are listed along with a brief explanation.

2.5.1 The Mass Conservation Equation

The equation for conservation of mass, or continuity equation, can be written as follows:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{V}) = 0$$

where ρ is the fluid density and \mathbf{V} is the velocity vector at a point in the flow field. The above equation is the continuity equation in the form of a partial differential equation. This equation relates the flow field variables at a point in the flow. It is a mathematical representation of the law of conservation of mass, i.e., mass can neither be created nor be destroyed.

The equation physically means that the time rate of change of volume of a moving fluid element of fixed mass, per unit volume of that element, is equal to the *divergence* ∇ of the fluid velocity vector \mathbf{V} . It should be noted here, that the fluid is assumed to be a continuum rather than a discrete medium.

In two-dimensions, the divergence of the velocity $\nabla \cdot \mathbf{V}$, is represented mathematically as follows,

$$\text{If } \mathbf{V} = V_x \mathbf{i} + V_y \mathbf{j} \equiv u \mathbf{i} + v \mathbf{j}$$

Then,

$$\nabla \cdot \mathbf{V} = \left(\frac{\partial}{\partial x} \mathbf{i} + \frac{\partial}{\partial y} \mathbf{j} \right) \cdot (u \mathbf{i} + v \mathbf{j}) = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}$$

2.5.2 Momentum Conservation Equations

For a two-dimensional unsteady viscous flow with density ρ , velocities (u, v) in Cartesian coordinates (x, y) , the Momentum Conservation Equations are given by the Navier-Stokes equations:

$$\rho \frac{\partial u}{\partial t} + \rho u \frac{\partial u}{\partial x} + \rho v \frac{\partial u}{\partial y} = -\frac{\partial p}{\partial x} + \frac{\partial}{\partial x} \left(\lambda \nabla \cdot \mathbf{V} + 2\mu \frac{\partial u}{\partial x} \right) + \frac{\partial}{\partial y} \left[\mu \left(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y} \right) \right]$$

$$\rho \frac{\partial v}{\partial t} + \rho u \frac{\partial v}{\partial x} + \rho v \frac{\partial v}{\partial y} = -\frac{\partial p}{\partial y} + \frac{\partial}{\partial x} \left[\mu \left(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y} \right) \right] + \frac{\partial}{\partial y} \left(\lambda \nabla \cdot \mathbf{V} + 2\mu \frac{\partial v}{\partial y} \right)$$

The approximate expression for λ is given as

$$\lambda = -\frac{2}{3}\mu$$

The above equations represent the complete Navier-Stokes equations for an unsteady, two-dimensional viscous flow. However to analyze the compressibility effects and the turbulence associated with high Reynolds number flows, some additional equations must be solved. These are the energy and transport equations for modeling the turbulence.

2.5.3 Viscous Energy Equation

The viscous energy equation is necessary in order to model the compressibility effects associated with high Reynolds number flows. Compressibility effects are encountered in gas flows at high velocity and/or in which there are large pressure variations. When the flow velocity approaches or exceeds the speed of sound of the gas or when the pressure change in the system is large, the variation of the gas density with pressure has a significant impact on the flow velocity, pressure, and temperature.

As the Mach number approaches 1.0 (which is referred to as the transonic flow regime), compressibility effects become important. The energy equation is derived using the first law of thermodynamics applied to an infinitesimal moving fluid element, and is mathematically formulated for a two-dimensional viscous flow as

$$\frac{\partial}{\partial t}(\rho E) = \rho \dot{q} + \frac{\partial}{\partial x} \left(k \frac{\partial T}{\partial x} \right) + \frac{\partial}{\partial y} \left(k \frac{\partial T}{\partial y} \right) - \nabla \cdot p\mathbf{V}$$

$$+ \frac{\partial(u\tau_{xx})}{\partial x} + \frac{\partial(u\tau_{yx})}{\partial y} + \frac{\partial(v\tau_{xy})}{\partial x} + \frac{\partial(v\tau_{yy})}{\partial y}$$

Here

- E is the total energy on a unit mass basis given by

$$E = h - \frac{p}{\rho} + \frac{\mathbf{V}^2}{2}$$

- h is the specific sensible enthalpy given as $h = c_p(T - T_{ref})$, $T_{ref}=298.15[\text{K}]$ and c_p is the specific heat at constant pressure.

- τ_{xx} and τ_{yy} are the normal stresses in the x and y directions given by

$$\tau_{xx} = \lambda(\nabla \cdot \mathbf{V}) + 2\mu \frac{\partial u}{\partial x}$$

$$\tau_{yy} = \lambda(\nabla \cdot \mathbf{V}) + 2\mu \frac{\partial v}{\partial y}$$

- τ_{xy} is the in-plane shear stress given as

$$\tau_{xy} = \tau_{yx} = \mu \left(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y} \right)$$

2.5.4 Turbulence Model

Most flows encountered in engineering practice are turbulent and therefore require different treatment. Turbulent flows are characterized by the following properties:

- Turbulent flows are highly unsteady,
- They are three-dimensional.
- They contain a great deal of vorticity.
- Turbulence increases the rate at which the conserved quantities are stirred.
- This brings fluids of different momentum content into contact.
- Turbulent flows contain coherent structures—repeatable and essentially deterministic events that are responsible for a large part of the mixing.
- Turbulent flows fluctuate on a broad range of length and time scales [29].

In order to include the effects of turbulence in any analysis, it is first necessary to have a model for the turbulence itself. The chief difficulty in modeling turbulent flows comes from the wide range of length and time scales associated with turbulent flow. As a result, turbulence models can be classified based on the range of these length and time scales that are modeled and the range of length and time scales that

are resolved. The more turbulent scales that are resolved, the finer the resolution of the simulation, and therefore the higher the computational cost. If a majority or all of the turbulent scales are modeled, the computational cost is very low, but the tradeoff comes in the form of decreased accuracy.

2.5.4.1 Reynolds-averaged Navier–Stokes Equations

One of the widely used approaches in modeling flow turbulence is the Reynolds-averaged Navier-Stokes (or RANS) equations. In Reynolds-averaged approaches to turbulence, all of the unsteadiness is averaged out i.e. all unsteadiness is regarded as part of the turbulence.

In a statistically steady flow, every variable can be written as the sum of a time-averaged value and a fluctuation about that value:

$$\phi(x_i, t) = \bar{\phi}(x_i) + \phi'(x_i, t)$$

This is known as the Reynolds Decomposition of the variable and forms ϕ the basis of RANS equations. The mean of the variable $\bar{\phi}$ can be expressed mathematically as

$$\bar{\phi}(x_i) = \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T \phi(x_i, t) dt$$

This expression is called the time average of the variable $\phi(x_i, t)$. Here t is the time and T is the averaging interval. This interval must be large compared to the typical scale fluctuations. If T is large enough, $\bar{\phi}$ does not depend on the time at which the averaging is started. It must be noted here that $\bar{\phi}$ is no longer a function of time and may vary only in space. Therefore all derivatives of $\bar{\phi}$ should mathematically be zeros. However, this is often (usually) ignored in modern treatments of RANS modeling.

Reynolds decomposition possesses the following two properties

$$\overline{\bar{\phi}} = \bar{\phi}, \text{ and } \overline{\phi'} = 0$$

The first relation is reflects one of the properties of Reynolds operator $R^2=R$. The second relation is the property of Reynolds decomposition, according to which the variable perturbations or fluctuations ϕ' are defined such that their time average equals zero.

Based on these properties the equations for mass and momentum conservations can be written as

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \bar{\mathbf{V}}) = 0$$

$$\rho \left[\bar{u} \frac{\partial \bar{u}}{\partial x} + \bar{v} \frac{\partial \bar{u}}{\partial y} \right] = -\frac{\partial \bar{p}}{\partial x} + \frac{\partial}{\partial x} \left(\lambda \nabla \cdot \bar{\mathbf{V}} + 2\mu \frac{\partial \bar{u}}{\partial x} \right) + \frac{\partial}{\partial y} \left[\mu \left(\frac{\partial \bar{v}}{\partial x} + \frac{\partial \bar{u}}{\partial y} \right) \right] - \rho \left[\frac{\partial \overline{u'^2}}{\partial x} + \frac{\partial \overline{u'v'}}{\partial y} \right]$$

$$\rho \left[\bar{u} \frac{\partial \bar{v}}{\partial x} + \bar{v} \frac{\partial \bar{v}}{\partial y} \right] = -\frac{\partial \bar{p}}{\partial y} + \frac{\partial}{\partial y} \left(\lambda \nabla \cdot \bar{\mathbf{V}} + 2\mu \frac{\partial \bar{v}}{\partial y} \right) + \frac{\partial}{\partial x} \left[\mu \left(\frac{\partial \bar{v}}{\partial x} + \frac{\partial \bar{u}}{\partial y} \right) \right] - \rho \left[\frac{\partial \overline{u'^2}}{\partial y} + \frac{\partial \overline{u'v'}}{\partial x} \right]$$

The above equations can be written in the tensor notation as

$$\rho \frac{\partial \overline{u_i u_j}}{\partial x_j} = -\frac{\partial \bar{p}}{\partial x_i} + \frac{\partial}{\partial x_i} \left(\lambda \nabla \cdot \bar{\mathbf{V}} + 2\mu \frac{\partial \bar{u}_i}{\partial x_i} \right) + \mu \frac{\partial^2 \bar{u}_i}{\partial x_i \partial x_j} - \rho \frac{\partial R_{ij}}{\partial x_i}, \quad i = 1, 2 \text{ (for 2-D)}$$

Here the notations can be interpreted as

$$(\overline{u_1}, \overline{u_2}) \equiv (\bar{u}, \bar{v}), \quad (\text{For the velocities})$$

$$(\overline{x_1}, \overline{x_2}) \equiv (\bar{x}, \bar{y}), \quad (\text{For the directions})$$

$$\overline{u_i u_j} = \begin{pmatrix} \overline{u_1^2} & \overline{u_1 u_2} \\ \overline{u_2 u_1} & \overline{u_2^2} \end{pmatrix} \equiv \begin{pmatrix} \overline{u^2} & \overline{uv} \\ \overline{vu} & \overline{v^2} \end{pmatrix}$$

R_{ij} is known as the Reynolds stress tensor given by

$$\rho R_{ij} \equiv \rho \overline{u'_i u'_j} = \rho \begin{pmatrix} \overline{u_1'^2} & \overline{u_1' u_2'} \\ \overline{u_2' u_1'} & \overline{u_2'^2} \end{pmatrix} \equiv \rho \begin{pmatrix} \overline{u'^2} & \overline{u'v'} \\ \overline{v'u'} & \overline{v'^2} \end{pmatrix}$$

As mentioned earlier, in modern RANS modeling of turbulent flows, it is a common practice to include time derivatives of the mean of the variables, the mass and momentum equations are formulated as

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \bar{\mathbf{V}}) = 0$$

$$\frac{\partial \rho \bar{u}_i}{\partial t} + \rho \frac{\partial \bar{u}_i \bar{u}_j}{\partial x_j} = -\frac{\partial \bar{p}}{\partial x_i} + \frac{\partial}{\partial x_i} \left(\lambda \nabla \cdot \bar{\mathbf{V}} + 2\mu \frac{\partial \bar{u}_i}{\partial x_i} \right) + \mu \frac{\partial^2 \bar{u}_i}{\partial x_i \partial x_j} - \rho \frac{\partial R_{ij}}{\partial x_i} \text{ For } i = 1, 2$$

The above equations are the RANS equations for unsteady flow in two dimensions. The details of the derivation, terminologies and Reynolds theories can be found in [30].

2.5.4.2 Boussinesq hypothesis

The Reynolds-averaged approach to turbulence modeling requires that the Reynolds stresses R_{ij} are appropriately modeled. A common method is to employ the Boussinesq hypothesis.

It was experimentally observed that turbulence decays unless there is shear in isothermal incompressible flows. Turbulence was found to increase as the mean rate of deformation increases. Boussinesq proposed in 1877 that the Reynolds stresses could be linked to the mean rate of deformation. In general, the in plane viscous shear stresses in two-dimensions are given by

$$\tau_{ij} = \mu e_{ij} = \mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right), \text{ for } i = 1, 2$$

On the same lines, linking Reynolds stresses to the mean rate of deformation (Boussinesq hypothesis), gives

$$\tau_{ij} = -\rho \overline{u'_i u'_j} = -\rho R_{ij} = \mu_t \left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i} \right)$$

Here μ_t is called the turbulent eddy viscosity. It is not homogeneous, i.e. it varies in space. However it is assumed to be isotropic, i.e. it is the same in all directions. This assumption is valid for many flows except for flows with strong separation or swirl. The turbulent viscosity is used to close the momentum equations and is determined using a turbulence model.

2.5.4.3 Spalart-Allmaras turbulence model

The momentum equations can be closed by solving a turbulence model equation. There are a number of turbulence models that are in use in modern turbulence analysis such as the Spalart-Allmaras model, $k - \varepsilon$ models and the $k - \omega$

models. Here the Spalart-Allmaras model will be used. This turbulence model, which is a one-equation model as opposed to the others, can be used to calculate the dynamic eddy viscosity, μ_t , by solving the one-equation transport model for $\tilde{\nu}$ which is identical to the turbulent kinematic viscosity. The transport equation is given by

$$\frac{\partial}{\partial t}(\rho\tilde{\nu}) + \frac{\partial}{\partial x_i}(\rho\tilde{\nu}u_i) = G_\nu + \frac{1}{\sigma_{\tilde{\nu}}} \left[\frac{\partial}{\partial x_j} \left\{ (\mu + \rho\tilde{\nu}) \frac{\partial \tilde{\nu}}{\partial x_j} \right\} + C_{b2}\rho \left(\frac{\partial \tilde{\nu}}{\partial x_j} \right)^2 \right] - Y_\nu$$

Here G_ν is the production of turbulent viscosity, and Y_ν is the destruction of turbulent viscosity that occurs in the near-wall region due to the wall blocking and viscous damping. $\sigma_{\tilde{\nu}}$ and C_{b2} are the constants and ν is the molecular kinematic viscosity[31].

The turbulent viscosity, μ_t , is computed from

$$\mu_t = \rho\tilde{\nu}f_{v1}$$

where the viscous damping function, f_{v1} , is given by

$$f_{v1} = \frac{X^3}{X^3 + C_{v1}^3} \quad \text{and} \quad X \equiv \frac{\tilde{\nu}}{\nu}, \quad \text{Here } C_{v1} \text{ is a constant.}$$

The turbulent viscosity production term G_ν is modeled as

$$G_\nu = C_{b1}\rho\tilde{S}\tilde{\nu}$$

where

$$\tilde{S} \equiv S + \frac{\tilde{\nu}}{\kappa^2 d^2} f_{v2} \quad \text{and} \quad f_{v2} = 1 - \frac{X}{1 + Xf_{v1}}$$

C_{b1} and κ are constants, d is the distance from the wall, and S is a scalar measure of the deformation tensor. S is based on the magnitude of the vorticity: $S \equiv \sqrt{2\Omega_{ij}\Omega_{ij}}$ where Ω_{ij} is the mean rate-of-rotation tensor and is defined by

$$\Omega_{ij} = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} - \frac{\partial u_j}{\partial x_i} \right)$$

The turbulent viscosity destruction term is modeled as

$$Y_\nu = C_{w1}\rho f_w \left(\frac{\tilde{\nu}}{d} \right)^2 \quad \text{where} \quad f_w = g \left[\frac{1 + C_{w3}^6}{g^6 + C_{w3}^6} \right]^{\frac{1}{6}}$$

$$g = r + C_{w2}(r^6 - r), \quad \text{where } r = \frac{\tilde{v}}{\tilde{S}\kappa^2 d^2}$$

Table 1 Model Constants

C_{b1}	0.1355
C_{b2}	0.622
$\sigma_{\tilde{v}}$	2/3
C_{v1}	7.1
C_{w1}	$\frac{C_{b1}}{\kappa^2} + \frac{(1 + C_{b2})}{\sigma_{\tilde{v}}}$
C_{w2}	0.3
C_{w3}	2.0
κ	0.4187

2.6 Computational Fluid Dynamics

Computational fluid dynamics (CFD) is the science of predicting fluid flow, heat transfer, mass transfer, chemical reactions, and related phenomena by solving the mathematical equations which govern these processes using a numerical process.

The governing equations, such as the mass, momentum and energy conservation equations, and the turbulence models, described in the previous section cannot be solved analytically except in special cases. Thus, the solution to these equations is obtained numerically. In order to approximate the solution, a *discretization method* is used which approximates the differential equations by a set of algebraic equations, which can then be solved simultaneously or iteratively.

Broadly, the strategy of CFD is to replace the continuous problem domain with a discrete domain using a grid. In the continuous domain, each flow variable is defined at every point in the domain. For instance, the pressure p in the continuous 1D domain shown in the figure below would be given as

$$p = p(x), 0 < x < 1$$

In the discrete domain, each flow variable is defined only at the grid points. So, in the discrete domain shown below, the pressure would be defined only at the N grid points. $p_i = p(x_i)$, $i = 1, 2, \dots, N$

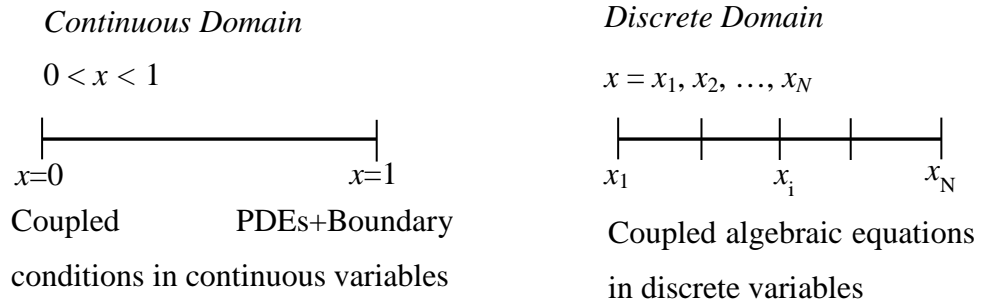


Figure 8 Continuous Domain and Discrete Domain

2.6.1 Discretization

In a CFD solution, relevant flow variables are solved only at the grid points. The values at other locations are determined by interpolating the values at the grid points. The governing partial differential equations and boundary conditions are defined in terms of the continuous variables p, \mathbf{V} etc in theoretical fluid dynamics, which are essentially functions of space and time. These can be approximate in the discrete domain in terms of the discrete variables p_i, \mathbf{V}_i etc. The discrete system is a large set of coupled, algebraic equations in terms of discrete variables.

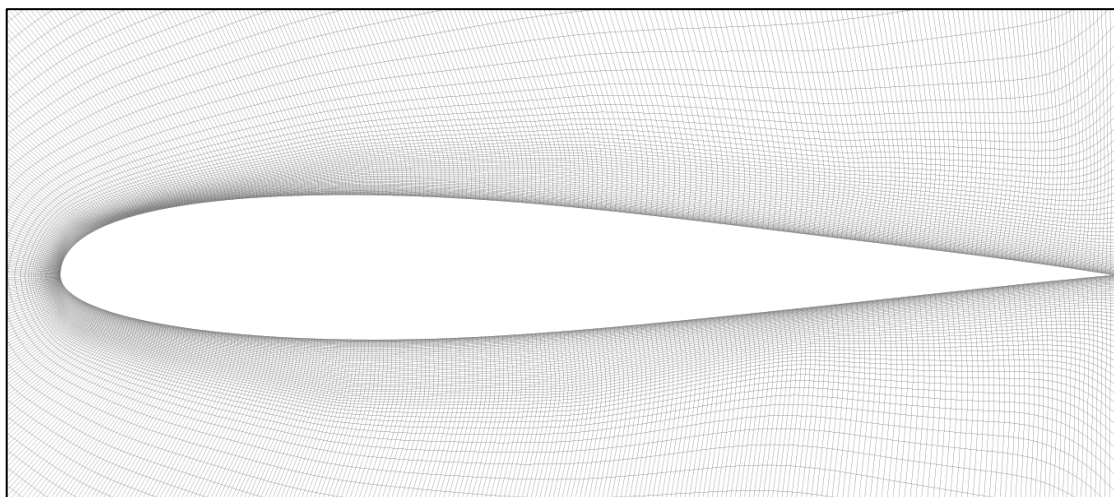


Figure 9 The Computational Spatial Grid

One of the many approaches to numerically solving the governing equations of flow is the Finite Volume method. This method forms the basis of almost 80 percent of present day flow solvers. In the finite-volume approach, the solution domain is subdivided into a finite number of small control volumes (cells) by a grid.

This sub-division is carried out using meshing algorithms. The grid defines the boundaries of the control volumes while the computational node lies at the center of the control volume. The *integral form* of the conservation equations are applied to the control volume defined by a cell to get the discrete equations for the cell. For example, the integral form of the continuity equation for steady, incompressible flow is

$$\oiint_S \rho \mathbf{V} \cdot d\mathbf{s} = 0$$

The integration is over the surface S of the control volume and $d\mathbf{s}$ is the elemental area vector. Physically, this equation means that the net volume flow into the control volume is zero. For any rectangular cell from the grid as shown below

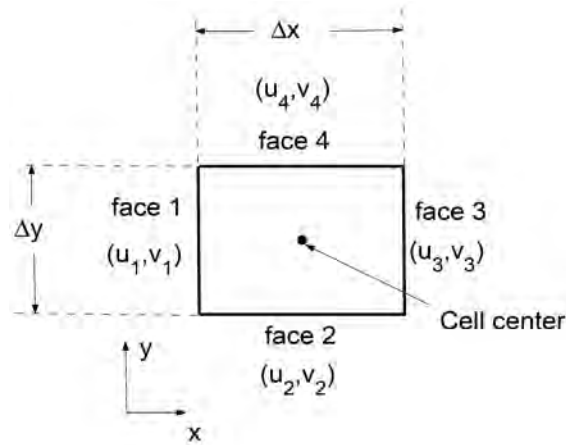


Figure 10 Typical finite volume cell

The velocity at face i is taken to be $\mathbf{V}_i = u_i \mathbf{i} + v_i \mathbf{j}$. Applying the mass conservation equation to the control volume defined by the cell gives

$$-u_1 \Delta y - v_2 \Delta x + u_3 \Delta y + v_4 \Delta x = 0$$

This is the discrete form of the continuity equation for the cell. It is equivalent to summing up the net mass flow into the control volume and setting it to zero. So it ensures that the net mass flow into the cell is zero i.e. that mass is conserved for the cell. Usually, though not always, the values at the cell centers are solved for directly by inverting the discrete system. The face values u_1, v_2 , etc. are obtained by suitably interpolating the cell-center values at adjacent cells.

In a general situation, the discrete equations are applied to the cells in the interior of the domain. For cells at or near the boundary, a combination of the discrete equations and boundary conditions would be applied. In the end, one would obtain a system of simultaneous algebraic equations with the number of equations being equal to the number of independent discrete variables. This set can then be solved to obtain the values of the variables at their respective grid points.

2.6.2 CFD based Shape Optimization

The focus of CFD applications has shifted from mere analysis to aerodynamic shape design. This shift has been mainly motivated by the availability of high performance computing platforms and by the development of new and efficient analysis and design algorithms. In particular automatic design procedures, which use CFD combined with gradient-based optimization techniques, have had a significant impact on the design process by removing difficulties in the decision making process faced by the aerodynamicist.

With recent research and code development efforts in the area of computational fluid dynamics, CFD has proven to be useful in supporting product design and development in many industrial applications. For many product designs where fluid flow simulations are needed, CFD analyses have proven to be quite useful in predicting the flow pattern for a given set of design parameters [32].

Aerodynamic shape optimization strategies integrated with CFD usually come into play at the preliminary design phase of an airplane. It is here that the conceptual design is refined from a shape optimization point of view. Modern shape optimization strategies usually involve the integration of a CFD code with an optimization algorithm. The CFD code performs the flow analysis on a specific shape and provides the optimization algorithm with values of the required components that make up the objective function. The optimization algorithm, based on the evaluation of the objective function value, perturbs the geometry in a direction of decreasing objective function gradients. The new geometry is then re-analyzed and the process is repeated till an optimum shape is reached.

3. ITERATIVE RESPONSE SURFACE BASED OPTIMIZATION

3.1 Introduction

The evaluation of aerospace designs is synonymous with the use of long running and computationally intensive simulations. Unlike the engineering methodologies adopted in the middle of the 20th century to design aerospace systems, in which predominantly hand calculations and wind tunnel tests were used in a cut-and-try fashion, engineers in the late 20th century have resorted to Computer Aided Engineering (CAE) of which CFD is an important part.

CFD, as outlined in the previous chapter, has evolved from a mere flow analysis tool to an important design tool. However, even with the ever expanding computational resources, it has widely been regarded as a computationally expensive platform, especially when it comes to very high-fidelity flow simulations. The use of long running expensive computer simulation in design, therefore leads to a fundamental problem when trying to compare and contrast various competing options— there are never sufficient resources to analyze all of the combinations of variables that one would wish [33].

This problem is particularly acute when using optimization schemes. All optimization methods depend on some form of internal model of the problem space they are exploring— for example a quasi-Newton scheme attempts to construct the Hessian at the current design point by sampling the design space. To build such a model when there are many variables can require large numbers of analyses to be carried out, particularly if using finite difference methods to evaluate gradients.

Objective function and constraints in aerodynamic shape optimization involving transonic flow numerical simulation, such as CFD, may be non-smooth and noisy. Non-smoothness is created by the presence of flow discontinuities such as shock waves. Noise can be caused either by the changes in computational mesh geometry due to free boundaries or by poor convergence of numerical schemes. Although these features can make a small change in some design parameters, it could lead to a huge ramification in the objective function or constraints.

These non-smoothness and noise issues of the objective function become more serious in gradient-based optimization methods (GBOMs), where the objective

function value as well as its gradient information is used. In multidisciplinary design optimization (MDO) problems, which usually have objective functions coupled with numerous constraints, it is significantly difficult to formulate the design problem with GBOMs. Because the optimization depends greatly on the formulation of the design problem, the process of searching for the optimum is likely to render just a local value. Another shortcoming of GBOMs is that because many analysis programs were not written with an automated design in mind, adaptation of these programs to an optimization code may need significant reprogramming in the analysis routine [34].

3.2 Response Surface Optimization

In statistics, *response surface methodology* explores the relationships between several explanatory variables and one or more response variables. The method was introduced by *G. E. P. Box* and *K. B. Wilson* in 1951. The main idea of response surface methodology is to use a sequence of designed experiments to obtain an optimal response. Incorporating this methodology in design optimization falls into the category of *Surrogate* or *Response Surface Optimization (RSO)*. It has been shown to be an effective approach for the design of computationally expensive models such as those found in aerospace systems, involving aerodynamics, structures, and propulsion, among other disciplines. Successful applications include the multidisciplinary optimal design of aerospike rocket nozzles, injectors and turbines for liquid rocket propulsion, and supersonic business aircrafts [35].

For a new or a computationally expensive design, optimization based on an inexpensive surrogate, such as *Response Surface Model* (also known as surrogate or approximation models), is a good choice. RSO allows for the determination of an optimum design, while at the same time providing insight into the workings of the design. A response model not only provides the benefit of low-cost for function evaluations, but it can also help revise the problem definition of a design task, which is not unusual for new efforts. Furthermore, it can conveniently handle the existence of multiple desirable design points and offer quantitative assessments of trade-offs as well as facilitate global sensitivity evaluations of the design variables[36].

Thus, the use of *Response Surface Models (RSM)* in optimization is becoming increasingly popular. The RSM is not in itself an optimizer, but rather a tool for increasing the speed of optimization. Instead of making direct calls to an expensive

numerical analysis code, such as CFD, an optimization routine takes values from a cheap surrogate model, that is formulated using a specific set of responses obtained from the numerical code. The popularity of such methods has probably increased due to the development of approximation methods which are better able to capture the nature of a multi-modal design space.

The main objective behind creating an RSM is to be able to predict the response of a system for an operating point without actually performing a simulated analysis at that point. The response of the system can then be predicted just by inputting the operating point values into the RSM and obtaining the value of the response. The RSM basically takes the shape of a mathematical equation $\hat{f}(\mathbf{x})$, essentially a quadratic polynomial, which takes the values of the design variables \mathbf{X} as an input, and returns an approximated value of the system response. Various optimization methodologies can then be employed to optimize this computationally cheap response model in order to obtain the best operating point. Some of the other benefits of using RSM include—

- It smoothens out the high-frequency noise of the objective function and is, thus, expected to find a solution near the global optimum.
- Various objectives and constraints can be attempted in the design process without additional numerical computations.
- It does not require a modification in analysis codes [1].

RSO is composed of four phases:

- *Sampling (Design of Experiments)*—this basically involves testing or obtaining actual values of the system response, by performing simulations for a select set of points within the design space.
- *Response Surface Construction*—based on the responses obtained for the sampling points, a RSM is constructed. The RSM is an approximation of the system response.
- *RSO*—Optimization algorithms are used to optimize the RSM and obtain the best operating point values of the system.
- *RSM improvement*—The RSM approximation is improved by training it further by including additional simulated responses.

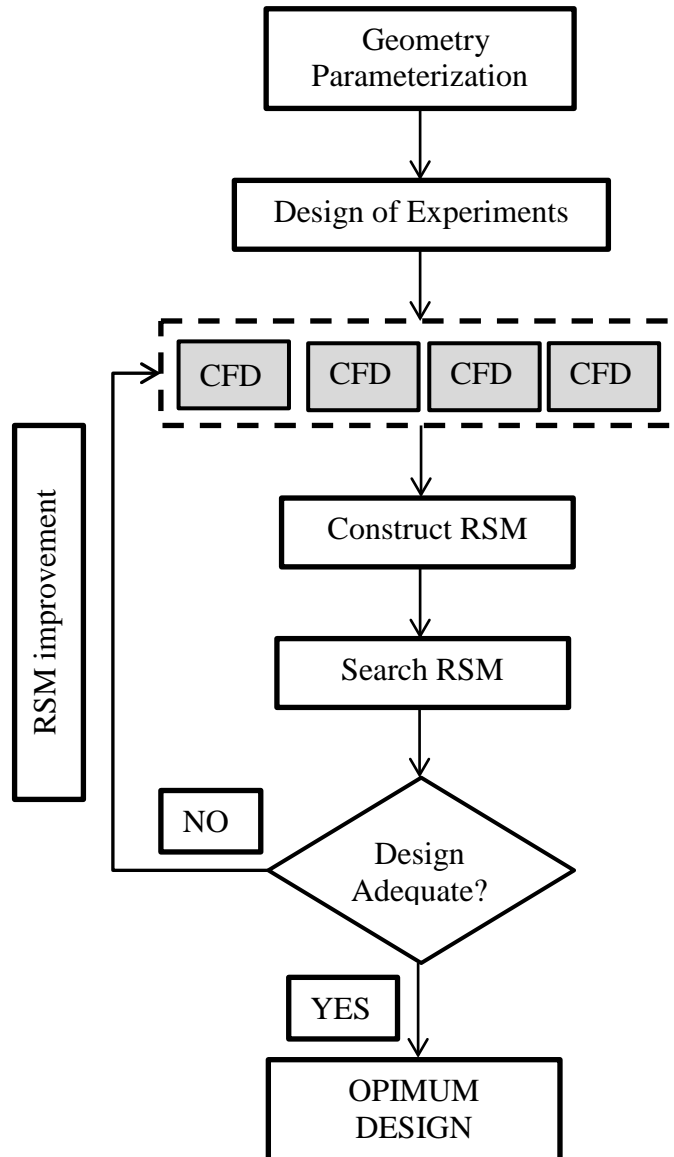


Figure 11 General RSO procedure

3.2.1 Design Space

From an aerodynamic shape optimization point of view, the *system* is basically the airfoil geometry that has to be optimized for a specific operating condition (airspeed and altitude). The *design points* are the design variables that completely define the airfoil geometry. As discussed in chapter 2, the design variables constitute the y-ordinates of the 10 B-spline control points that control the shape of the upper and lower surfaces of the airfoil, and the angle of attack.

The design space is the region bounded by the upper and lower limits of the design variables. This implies that the design variables are allowed to vary only

within the limits defined by the design space. It is defined such that, overly unusual or unrealistic shapes are not attained. It is also dependent on the structural and material constraints. For example, for any variant of the airfoil shape, the skin should not deform beyond the limits of elasticity, or the thickness should not be less than the required minimum thickness in-order to have enough room for the fuel tank etc.

3.2.2 Design of Experiments

In the Design of Experiments (DoE) phase the design space is systematically explored using a technique, which generates the test matrix of design points to be probed in each computational experiment. The objective of this task is to fragment the design space in a specific format such that a matrix of design variable values is obtained. This is done by discretizing the variation range of each design variable into N_s levels. Combining the values of all the design variables at a specific level yields one experiment. Combining all the experiments therefore forms a set of N_s experiments, which is thereby referred to as a DoE.

If \mathbf{X} is the design vector consisting of N_{var} design variables (DV), and if each design variable is split into N_s levels, the DoE matrix is given by,

$$\mathbf{X}_{DoE} = \begin{array}{c} \left[\begin{array}{cccc} \mathbf{x}_{11} & \mathbf{x}_{12} & \cdots & \mathbf{x}_{1N_{\text{var}}} \\ \mathbf{x}_{21} & \mathbf{x}_{22} & \cdots & \mathbf{x}_{2N_{\text{var}}} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{x}_{N_s1} & \mathbf{x}_{N_s2} & \cdots & \mathbf{x}_{N_sN_{\text{var}}} \end{array} \right] \begin{array}{l} \leftarrow \text{Experiment} - 1 \\ \leftarrow \text{Experiment} - 2 \\ \vdots \\ \leftarrow \text{Experiment} - N_s \end{array} \\ \underbrace{\hspace{10em}}_{\text{Design Variables DV 1 to DV } N_{\text{var}}} \end{array}$$

3.2.2.1 Latin Hypercube Sampling

Latin Hypercube Sampling (LHS) is a statistical method for generating a distribution of plausible collections of design variable values from a multidimensional distribution. This method is often used as a DoE technique.

In geometry, a *hypercube* is an n -dimensional analogue of a square ($n = 2$) and a cube ($n = 3$). It is a closed, compact, convex figure whose 1-skeleton consists of groups of opposite parallel segments aligned in each of the space's dimensions, perpendicular to each other and of the same length. A *hyperplane* is also a concept which is a generalization of the plane into a different number of dimensions. A hyperplane of an n -dimensional space is a flat subset with dimension $n - 1$.

In the context of statistical sampling, a square grid containing sample positions is a Latin square if (and only if) there is only one sample in each row and each column. A Latin Hypercube is the generalization of this concept to an arbitrary number of dimensions, whereby each sample is the only one in each axis-aligned hyperplane containing it.

When sampling a function of N_{var} variables, the range of each variable is divided into equally probable intervals. The N_s sample points are then placed to satisfy the Latin Hypercube requirements; doing so forces the number of divisions, N_s , to be equal for each variable. It should be noted that this sampling scheme does not require more samples for more dimensions (variables); this independence is one of the main advantages of this sampling scheme. Another advantage is that random samples can be taken one at a time, remembering which samples were taken so far. For example, for $N_{var} = 4$ (4 design variables), and $N_s = 4$ (4 levels), a Latin Hypercube Sampling may take the form:

2	1	3	4	← Level – 1
3	2	4	1	← Level – 2
1	4	2	3	← Level – 3
4	3	1	2	← Level – 4
↑ DV 1	↑ DV 2	↑ DV 3	↑ DV 4	

Building a *Latin hypercube*, that is the multidimensional, can be done in a similar way. The design space of each dimension is split into equal number of levels and the points are placed in the levels such that any arbitrary vector emerging from the points in a direction parallel to any of the dimensional axes does not encounter with any other point in its way.

This is achieved using the following technique. If \mathbf{X} denotes the $N_s \times N_{var}$ the DoE matrix N_s points in N_{var} dimensions (each row represents a point), then each column of \mathbf{X} is filled with random permutations $(1, 2, \dots, N_s)$ and stratified such that no specific point in any row is repeated in more than one column. This set is then normalized such that values lie within $[0,1]^{N_{var}}$.

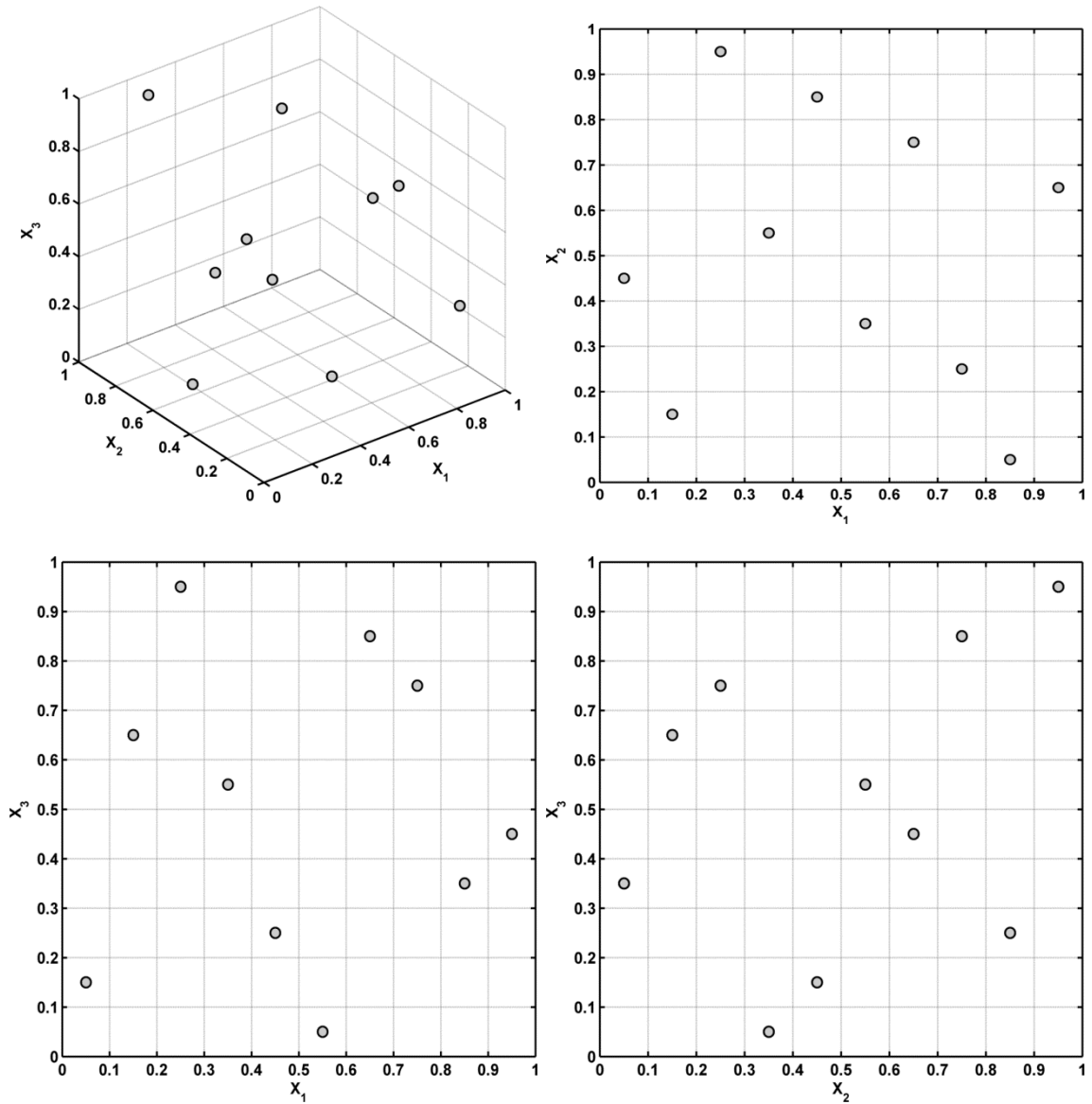


Figure 12 Three-variable, ten-point Latin hypercube sampling plan shown in three dimensions (top left), along with its two-dimensional projections.

3.2.2.2 Space Filling LHS

One of the most widely-used measures to evaluate the uniformity (*space-fillingness*) of a sampling plan is the *maximin* metric introduced by Johnson *et al.* (1990). The criterion based on this may be defined as follows—

Let d_1, d_2, \dots, d_m be the list of the unique values of distances between all possible pairs of points in a DoE \mathbf{X} , sorted in the ascending order. Here the distance d_j is basically defined by the p-norm of the space given by:

$$d_p(\mathbf{x}^{(i_1)}, \mathbf{x}^{(i_2)}) = \left(\sum_{j=1}^k |\mathbf{x}^{(i_1)} - \mathbf{x}^{(i_2)}|^p \right)^{1/p}$$

For $p = 1$ this is the rectangular distance and $p = 2$ yields the *Euclidean norm*. There is little evidence in the literature of one being more suitable than the other for sampling plan evaluation if no assumptions are made regarding the structure of the model to be fitted, though it must be noted that the rectangular distance is considerably cheaper to compute.

Further, let J_1, J_2, \dots, J_m be defined such that J_j is the number of pairs of points in \mathbf{X} separated by the distance d_j —Then \mathbf{X} is called a *maximin* plan among all available plans if it maximizes d_j and, among plans for which this is true, minimizes J_j [37].

3.2.2.3 Design Matrix

The design matrix is formed by concatenating the values of the design variables at all levels. In order to do so, the design space needs to be discretized into levels which are equal to the desired number of computer simulations to be performed. The design space as described above is the region bounded by the upper and lower limits of the design variables. These are the y-ordinates of the 10 control points (five control points for each surface) and the angle of attack. CPs 2 through 6 control the upper surface while CPs 9 through 13 control the lower surface of the airfoil.

For this, the lower and upper limits for each of the 10 control points and the angle of attack are defined. The lower limit of the control points is taken to be 75% of the baseline values and the upper limit is taken to be 25% above the baseline values. The lower and upper limits of the angle of attack are case dependent varying between 0 and 4°.

The range of each of the design variables DV_{Range} (the design space) is the difference between the upper, DV_{Upper} , and lower limits, DV_{Lower} , of the design variable. This range is discretized into equal number of levels N_s which is equivalent to the number of experiments (computer simulations) to be performed. To obtain the values of the design variables at each level, first a LHS plan is generated for the 11

design variables and N_s levels. This generates a matrix L of size $(N_s \times 11)$, with the N_s values in each of the 11 columns varying from 0 to 1 in a LHS pattern.

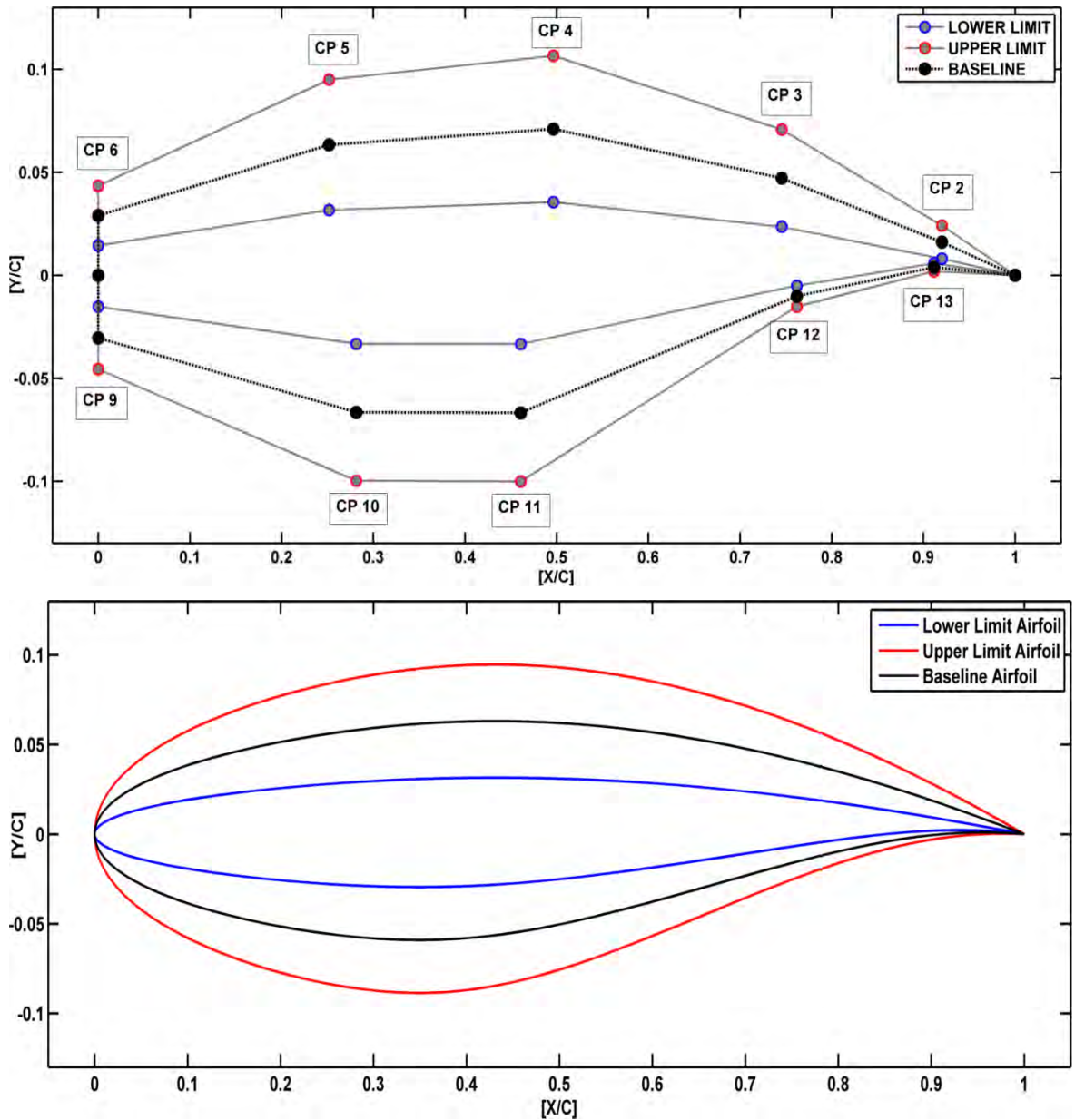


Figure 13 Lower Limit, Upper limit and Baseline Control Points and Airfoil shapes

The values of the design variables at each level are then obtained based on the following equation:

$$\mathbf{X}_{DoE}(i, j) = DV_{Lower}(j) + \left(DV_{Range}(j) \times L(i, j) \right), \quad \text{For } i = 1, 2, \dots, N_s$$

$$\text{For } j = 1, 2, \dots, 11$$

The matrix thus formed, describes the set of airfoil geometries for which the CFD simulations are to be performed in order to construct the RSM.

3.2.3 Constructing the RSM

RSM builds a response model by calculating data points with experimental design theory to prescribe a response of a system with independent variables. The relationship can be written in a general form as follows:

$$y = F(\mathbf{X}) + \varepsilon$$

where ε represents the total error, which is often assumed to have a normal distribution with a zero mean. Consider a sampling plan \mathbf{X} and a set of N_s observed values comprising the responses obtained from the computer simulations:

$$\mathbf{X} = \mathbf{X}_{DoE} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1N_{var}} \\ x_{21} & x_{22} & \cdots & x_{2N_{var}} \\ \vdots & \vdots & \ddots & \vdots \\ x_{N_s1} & x_{N_s2} & \cdots & x_{N_sN_{var}} \end{bmatrix} \rightarrow \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \\ \vdots \\ \mathbf{y}_{N_s} \end{bmatrix}$$

The polynomial approximation of order m (degree $m - 1$) of an underlying function f is, essentially, a Taylor series expansion of f truncated after $m - 1$ terms. This suggests that a higher order expansion will usually yield a more accurate approximation. However, the greater the number of terms, the more flexible the model becomes and there is a danger of over-fitting the noise that may be corrupting the underlying response. For this reason, the order of the polynomial has been restricted to 3.

A full quadratic polynomial (degree 2, order 3) approximation of F can be written as:

$$\hat{\mathbf{y}} = \hat{f}(\mathbf{x}, \boldsymbol{\beta}) = \beta_1 + \sum_{i=1}^{N_{var}} \beta_i x_i + \sum_{j=1}^{N_{var}} \beta_{jj} x_j^2 + \sum_{i=1}^{N_{var}-1} \sum_{j=i+1}^{N_{var}} \beta_{ij} x_i x_j$$

Here $\beta_0, \beta_i, \beta_{ij}$ etc. are the regression coefficients of the polynomial. The total number of these coefficients is $n_t = (N_{var} + 1)(N_{var} + 2)/2$. These values can be determined using the standard least-square fitting regression of an over determined problem:

$$\mathbf{y} = \Phi \boldsymbol{\beta}$$

Here \mathbf{y} is the initial response matrix $[y_1, y_2, \dots, y_{N_s}]^T$ and Φ is the Vandermonde matrix of size $(N_s \times N_{\text{var}})$ given by:

$\Phi =$

$$\begin{bmatrix} 1 & x_{11} & x_{12} & \dots & x_{1N_{\text{var}}} & x_{11}^2 & x_{12}^2 & \dots & x_{1N_{\text{var}}}^2 & x_{11}x_{12} & x_{11}x_{13} & \dots & x_{1N_{\text{var}}-1}x_{1N_{\text{var}}} \\ 1 & x_{21} & x_{22} & \dots & x_{2N_{\text{var}}} & x_{21}^2 & x_{22}^2 & \dots & x_{2N_{\text{var}}}^2 & x_{21}x_{22} & x_{21}x_{23} & \dots & x_{2N_{\text{var}}-1}x_{2N_{\text{var}}} \\ \vdots & \vdots & \vdots & \dots & \vdots & \vdots & \vdots & \dots & \vdots & \vdots & \vdots & \dots & \vdots \\ 1 & x_{N_s1} & x_{N_s2} & \dots & x_{N_sN_{\text{var}}} & x_{N_s1}^2 & x_{N_s2}^2 & \dots & x_{N_sN_{\text{var}}}^2 & x_{N_s1}x_{N_s2} & x_{N_s1}x_{N_s3} & \dots & x_{N_sN_{\text{var}}-1}x_{N_sN_{\text{var}}} \end{bmatrix}$$

$$\text{And } \boldsymbol{\beta} = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_{n_t} \end{bmatrix}$$

3.2.3.1 Linear Least-Squares Solution

The method of least squares is a standard approach to the approximate solution of over determined systems, i.e., sets of equations in which there are more equations than unknowns. Least-squares means that the overall solution minimizes the sum of the squares of the errors made in the results of every single equation.

Such a system, as described above, usually has no solution, so the goal is instead to find the coefficients $\boldsymbol{\beta}$ which fit the equations best, in the sense of solving the quadratic minimization problem:

$$\hat{\boldsymbol{\beta}} = \min_{\boldsymbol{\beta}} S(\boldsymbol{\beta})$$

Here the objective function S is given by

$$S(\boldsymbol{\beta}) = \sum_{i=1}^{N_s} \left| y_i - \sum_{j=1}^{n_t} \Phi_{ij} \beta_j \right|^2 = \|\mathbf{y} - \Phi \boldsymbol{\beta}\|^2$$

Considering the i^{th} residual to be

$$r_i = y_i - \sum_{j=1}^{n_t} \Phi_{ij} \beta_j$$

Then S can be written as

$$S = \sum_{i=1}^{N_s} r_i^2.$$

S is minimized when the gradient vector is zero. The elements of the gradient vector are the partial derivatives of S with respect to the parameters.

$$\frac{\partial S}{\partial \beta_j} = 2 \sum_{i=1}^{N_s} r_i \frac{\partial r_i}{\partial \beta_j}, \quad \text{for } (j = 1, 2, \dots, n_t)$$

$$\text{And } \frac{\partial r_i}{\partial \beta_j} = -\Phi_{ij}$$

Substitution of the expressions for the residuals and the derivatives into the gradient equations gives

$$\frac{\partial S}{\partial \beta_j} = 2 \sum_{i=1}^{N_s} \left(y_i - \sum_{k=1}^{n_t} X_{ik} \beta_k \right) (-X_{ij}), \quad \text{for } (j = 1, 2, \dots, n_t)$$

Thus, if $\hat{\beta}$ minimizes S , then

$$2 \sum_{i=1}^{N_s} \left(y_i - \sum_{k=1}^{n_t} \Phi_{ik} \hat{\beta}_k \right) (-\Phi_{ij}) = 0, \quad \text{for } (j = 1, 2, \dots, n_t)$$

Upon rearrangement,

$$\sum_{i=1}^{N_s} \sum_{k=1}^{n_t} \Phi_{ij} \Phi_{ik} \hat{\beta}_k = \sum_{k=1}^{n_t} \Phi_{ij} y_i, \quad \text{for } (j = 1, 2, \dots, n_t)$$

The normal equations are written in the matrix notation as

$$(\Phi^T \Phi) \hat{\beta} = \Phi^T \mathbf{y},$$

$$\text{or } \hat{\beta} = (\Phi^T \Phi)^{-1} \Phi^T \mathbf{y}$$

The solution of the normal equations yields the vector $\hat{\beta}$ of the optimal parameter values. It must be noted that in order to construct a sufficiently trained RSM, the number of samples (or levels) N_s must be 1.5~3 times the number of regression coefficients, n_t .

3.2.3.2 Model Testing

Once the RSM is available, it is imperative to establish the predictive capabilities of the surrogate model away from the available data. In the context of an RSA, several measures of predictive capability are available

Adjusted root mean square error

The error ε_i at any point is i is given by

$$\varepsilon_i = y_i - \hat{y}_i$$

where y_i is the actual value and \hat{y}_i is the predicted value. Hence the adjusted root mean square (RMS_{adj}) error σ_a is given by

$$\sigma_a = \sqrt{\frac{\sum_{i=1}^{N_s} \varepsilon_i^2}{(N_s - n_t)}}$$

For a good fit, σ_a should be small compared to the data.

Coefficient of multiple determination

The adjusted coefficient of multiple determination R_{adj}^2 defines the prediction capability of the RSM as

$$R_{adj}^2 = 1 - \left(\frac{\sigma_a^2 (N_s - 1)}{\sum_{i=1}^{N_s} (y_i - \bar{y})^2} \right), \quad \text{where } \bar{y} = \frac{\sum_{i=1}^{N_s} y_i}{N_s}$$

For a good fit, R_{adj}^2 should be close to 1.

3.2.3.3 The RSM

From the aerodynamic shape optimization perspective with 11 design variables ($N_{\text{var}} = 11$), the number of coefficients are $n_t = 78$. A typical RSM would take the form—

$$\hat{y} \equiv \hat{f}(\mathbf{x}, \boldsymbol{\beta}) =$$

$$\begin{aligned} & \beta_1 + \beta_2 x_1 + \beta_3 x_2 + \beta_4 x_3 + \beta_5 x_4 + \beta_6 x_5 + \beta_7 x_6 + \beta_8 x_7 + \beta_9 x_8 + \beta_{10} x_9 + \beta_{11} x_{10} + \\ & \beta_{12} x_{11} + \beta_{13} x_1^2 + \beta_{14} x_2^2 + \beta_{15} x_3^2 + \beta_{16} x_4^2 + \beta_{17} x_5^2 + \beta_{18} x_6^2 + \beta_{19} x_7^2 + \\ & \beta_{20} x_8^2 + \beta_{21} x_9^2 + \beta_{22} x_{10}^2 + \beta_{23} x_{11}^2 + \\ & \beta_{24} x_1 x_2 + \beta_{25} x_1 x_3 + \beta_{26} x_1 x_4 + \beta_{27} x_1 x_5 + \beta_{28} x_1 x_6 + \beta_{29} x_1 x_7 + \beta_{30} x_1 x_8 + \\ & \beta_{31} x_1 x_9 + \beta_{32} x_1 x_{10} + \beta_{33} x_1 x_{11} + \\ & \beta_{34} x_2 x_3 + \beta_{35} x_2 x_4 + \beta_{36} x_2 x_5 + \beta_{37} x_2 x_6 + \beta_{38} x_2 x_7 + \beta_{39} x_2 x_8 + \beta_{40} x_2 x_9 \\ & + \beta_{41} x_2 x_{10} + \beta_{42} x_2 x_{11} + \\ & \beta_{43} x_3 x_4 + \beta_{44} x_3 x_5 + \beta_{45} x_3 x_6 + \beta_{46} x_3 x_7 + \beta_{47} x_3 x_8 + \beta_{48} x_3 x_9 + \beta_{49} x_3 x_{10} \\ & + \beta_{50} x_3 x_{11} \\ & + \beta_{51} x_4 x_5 + \beta_{52} x_4 x_6 + \beta_{53} x_4 x_7 + \beta_{54} x_4 x_8 + \beta_{55} x_4 x_9 + \beta_{56} x_4 x_{10} + \beta_{57} x_4 x_{11} + \dots \end{aligned}$$

$$\begin{aligned}
& \beta_{58}x_5x_6 + \beta_{59}x_5x_7 + \beta_{60}x_5x_8 + \beta_{61}x_5x_9 + \beta_{62}x_5x_{10} + \beta_{63}x_5x_{11} + \\
& \beta_{64}x_6x_7 + \beta_{65}x_6x_8 + \beta_{66}x_6x_9 + \beta_{67}x_6x_{10} + \beta_{68}x_6x_{11} + \\
& \beta_{69}x_7x_8 + \beta_{70}x_7x_9 + \beta_{71}x_7x_{10} + \beta_{72}x_7x_{11} + \\
& \beta_{73}x_8x_9 + \beta_{74}x_8x_{10} + \beta_{75}x_8x_{11} + \\
& \beta_{76}x_9x_{10} + \beta_{77}x_9x_{11} + \\
& \beta_{78}x_{10}x_{11}
\end{aligned}$$

3.2.4 Optimizing the RSM

Optimizing the RSM is the third step in the response surface optimization process. The RSM obtained, $\hat{y} \equiv \hat{f}(\mathbf{X}, \boldsymbol{\beta})$, as described earlier is a quadratic polynomial consisting of n_t terms in N_{var} dimensions (design variables). It is basically an algebraic equation which takes the values of the design variables (\mathbf{X}) and parameters ($\boldsymbol{\beta}$) as the input and returns a scalar value which is an approximation of the system's response.

Also, as the RSM is a quadratic polynomial, it is expected to be a smooth continuous function in terms of the design variables \mathbf{X} . Therefore there are no issues relating to noise and non-smoothness that are associated with direct optimization method. Optimizing the RSM is therefore simpler, computationally less expensive and efficient when compared to direct optimization of the system (i.e. optimizing using the gradient values of the system response).

The optimization is carried on the RSM using a Sequential Quadratic Programming (SQP) algorithm. This algorithm has been described at length in chapter 2. The values of the design variables obtained at the end of the SQP implementation, (\mathbf{X}^*), are the values that minimize the RSM. These are the optimum values of the design variables that minimize the objective function.

3.2.5 System Response

The system response, a term which has been excessively used in the descriptions above, is the scalar value of the objective functions that are defined for the purpose of aerodynamic shape optimization.

The objective function used here is the Lift-Moment Constrained Drag minimization problem given by

$$J = \begin{cases} \omega_L \left(1 - \frac{C_L}{C_L^*}\right)^2 + \omega_M \left(1 - \frac{C_M}{C_M^*}\right)^2 + \omega_D \left(1 - \frac{C_D}{C_D^*}\right)^2 & \text{if } C_D > C_D^* \\ \omega_L \left(1 - \frac{C_L}{C_L^*}\right)^2 + \omega_M \left(1 - \frac{C_M}{C_M^*}\right)^2 & \text{otherwise} \end{cases}$$

Here $\omega_L, \omega_M, \omega_D$ are user defined weights, and C_L^*, C_M^*, C_D^* are the target (required) lift, moment and drag coefficients respectively. This function has been described at length in Chapter 2 (section 2.4). The above function is evaluated using the results obtained from the CFD solver. The solver performs the flow simulation for an airfoil shape that is obtained from the set of design variables (Control point coordinates and angle of attack) for a particular operating condition (Mach number, operating pressure, temperature).

At the end of the simulation, the CFD solver yields the values of C_L, C_M and C_D for the shape being investigated. These values are then used to evaluate the objective function as above. This value of J is considered as a particular response of the system at a specific operating point (in this case, the operating point is the above airfoil shape governed by specific values of the design variables). A set consisting of multiple values of such operating points, in conjunction with the corresponding values of their responses, is used to construct the RSM as described in the previous sections.

3.3 Iterative Improvement of the RSM

In the standard RSO scheme the design space is normally explored using a space filling DoE technique. The range of variation of the design variables is chosen about a reference design, considering a large portion of the design space. On these selected points, the objective functions are evaluated using the CFD solver and the information is used to build a RSM based on a least-square-fitting quadratic polynomial. An optimization problem based on the RSM is then solved using a gradient method, such as SQP as described above, to find the minimum.

Because these approximations carry a bias error, the minimum thus found needs to be validated. Therefore an additional CFD run is performed at the end of the process to verify the performance of the optimum design obtained by solving the equivalent optimization problem.

To improve the prediction capability of the RSM, an iterative RSO process can be employed. This process can be summarized as follows—

- 1) An initial optimum is found solving the constrained minimization problem based on the least-square fitting quadratic polynomials of the objective and constraint functions, built upon a large design space.
- 2) The actual value of the objective function is computed for the optimum design found. This computation is done based on the values obtained from a CFD solver run for the optimum airfoil shape found in step 1.
- 3) The relevant geometry values are added to the initial set of samples (the initial set used to construct the RSM). The actual value of the objective function (found from step 2) is added to the set of responses.
- 4) Least-square fitting and minimization are performed again on the basis of the new set of data.
- 5) Steps 2 to 4 are repeated until the minimum computed from the response surface optimization and the function evaluation does not vary within a specified tolerance [38].

When no further cost function reduction is obtained, an additional step can be performed. The bounds of the design space of the SQP optimization based on the polynomial approximation can be changed, by updating the center point with the position of the last optimum found, while maintaining its size. To approximate the actual function in this new region of the design space, the polynomial computed as the least-square fit of all the previous design points are used and iteratively corrected about the current minimum in the same way as it is done previously.

3.3.1 Algorithm Flow Chart

The algorithm begins with an initial airfoil geometry that has to be optimized for a specific operating condition (airspeed and altitude). This geometry is parameterized using B-splines curve fitting (as described in chapter 2) to obtain values of the design variables ($N_{var} = 11 \rightarrow 10$ control points and the angle of attack). The range of variation for each of the design variables is defined by assigning an upper and a lower limit. These limits, which are based on certain structural constraints (as described earlier), become the design space for the optimization problem.

The design space (design variables) is then discretized into a number of levels (N_s) which is equal to the number of CFD simulations to be performed. The N_s values of each of the design variables are then arranged in a set, \mathbf{X}_{DoE} as per the LHS requirements.

CFD simulation is performed for each of the samples in the DoE for a specific operating condition (airspeed and altitude). The results (C_L, C_D, C_M, C_{Px_s}) are recorded and using these values, the objective function is evaluated. This process is carried out for all the samples and a results array, $[\mathbf{y}]$, is formed wherein the objective function values are arranged corresponding to their respective samples as in the DoE.

An over-determined problem is formulated based on the DoE and the responses (objective function values)— $[\Phi][\beta] = [\mathbf{y}]$ ($[\Phi]$ is the Vandermonde matrix in terms of the design variables \mathbf{x}). This equation is then solved for $[\beta]$, which is a matrix consisting of the polynomial regression coefficients, in a linear least-squares sense. These values of $[\beta]$ are then used to construct the RSM, $\hat{\mathbf{y}} \equiv \hat{f}(\mathbf{x}, \beta)$, which is a quadratic polynomial with $n_t = 78$ terms.

The RSM is then optimized using the SQP technique and a set of optimum values for the design variables, $[\mathbf{x}_{OPT}]$, and the corresponding function value, J_{appr} , is obtained.

The optimum value needs to be verified, and so the airfoil geometry corresponding to $[\mathbf{x}_{OPT}]$ is analyzed by performing an additional CFD simulation. The objective function value, using the results of this simulation, is computed J_{actual} . If the difference between J_{appr} and J_{actual} lies within a certain tolerance value, then $[\mathbf{x}_{OPT}]$ is the minimum.

Else, $[\mathbf{x}_{OPT}]$ is added to the initial sampling set \mathbf{X}_{DoE} and the corresponding actual function value, J_{actual} , is added to the results array, $[\mathbf{y}]$. The over-determined problem $[\Phi][\beta] = [\mathbf{y}]$ is reformulated and solved for $[\beta]$. $\hat{\mathbf{y}}$ is then re-constructed with the updated values of $[\beta]$, and optimized to get a new set of optimum values $[\mathbf{x}_{OPT}]$. This process of reconstructing the RSM with updated values of $[\beta]$ is repeated till the difference between the actual and approximated objective function value does not within a specified tolerance.

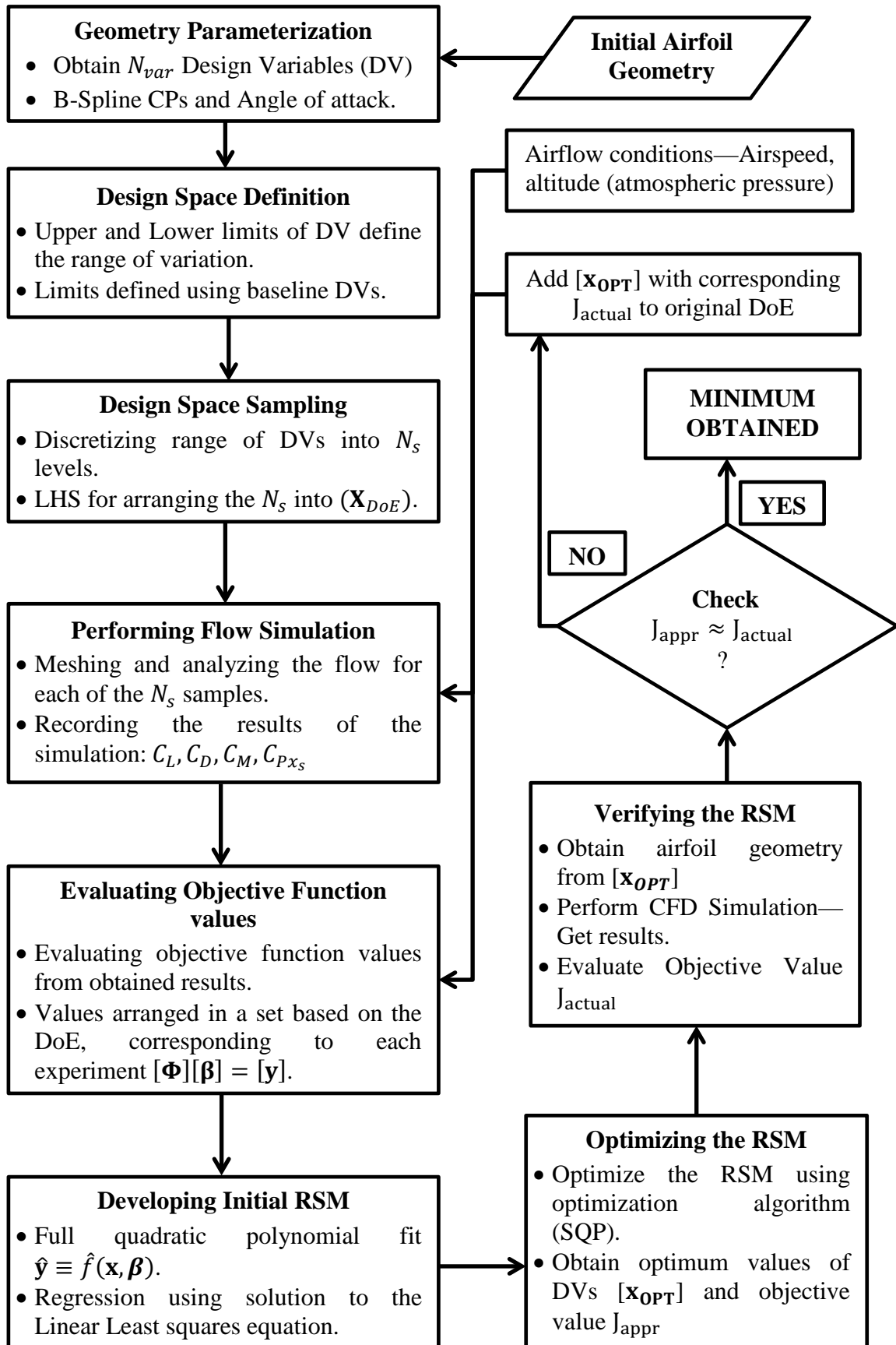


Figure 14 Iterative RSO process flow chart

3.4 MATLAB Implementation of the Iterative RSO

The Iterative RSO method for the airfoil aerodynamic shape optimization problem is implemented using MATLAB®, by integrating it with GridPro (meshing software) and ANSYS® FLUENT (flow solver). This implementation using MATLAB consists of two main blocks—CFD Block and the OPTIMIZATION Block.

The CFD Block, as the name implies, consists of all the processes that are required to perform a single CFD simulation. These include the meshing, flow solving and results generation processes. For a single CFD simulation, the following steps are performed via the CFD block—

1. Airfoil coordinate files generation—In this step two files are created that consist the x, y and z coordinates of the upper and lower surface of the airfoil (GPro_af1.DAT and GPro_af2.DAT). These files are required for the mesh generation process.
2. GridPro Topology Input Language (TIL) file—A TIL file is generated that consists of the commands that need to be executed by GridPro in-order to generate the mesh (FINAL3.FRA). This file is airfoil geometry specific in that the topology is constructed around the airfoil geometry (i.e. the topology varies with different airfoil shapes).
3. Trigger GridPro—MATLAB triggers the GridPro engine (GGrid) that takes in the coordinate files and the TIL file as the input, performs the meshing process and returns the mesh file in the required format (FLO_MESH.msh).
4. ANSYS-FLUENT variable file—Here a variables file is generated that includes various values pertaining to the operating condition of the flow. These include the operating pressure (P), ambient temperature (T), Mach number (M). In addition, the angle of attack in terms of the unit vector magnitudes in the horizontal and vertical direction of the flow is included (OP.var).
5. Trigger ANSYS-FLUENT—MATLAB triggers ANSYS-FLUENT which reads the generated mesh file, loads the variable file, and performs the

flow simulation by executing the commands included in the journal file (FLUENT_RUN3.JOU). This is discussed in the subsequent sections.

6. Result Files—At the end of the simulation, the various results are saved in individual files. The results include—lift and drag forces, moments, surface pressure coefficient distribution and surface shear distributions. (LIFT.CSV, DRAG.CSV, MOMENT.CSV, CP_TOP.CSV, CP_LOW.CSV, Stress_TOP.CSV and Stress_LOW.CSV).
7. These files are read into MATLAB and the values obtained from these files are fed into the OPTIMIZATION Block.

The OPTIMIZATION Block, consists of all the processes that are required for carrying out the various steps in the iterative RSO methodology. These include geometry parameterization, RSM construction and optimization and the iterative RSO methodology. The two blocks are interconnected, i.e. data flows to and fro as required. The following steps are performed via the OPTIMIZATION Block—

1. Airfoil parameterization—In this step, the parameterization of the airfoil is performed using B-spline curve fitting. Here the control point coordinates and a curve-parameter set is returned. The y-ordinates of the control points (along with the angle of attack) are used as design variables (DV) to perform the optimization process. (optimum_fit.m)
2. DV discretization—A range for the DVs obtained in the above step is defined in terms of lower and upper limits. The range of each of the DVs is then discretized into equal number of levels and arranged in a LHS plan and DoE matrix based on this plan is obtained. (LH_Doe.m, Sampling.m)
3. Objective function evaluation—For each of the planned experiments in the DoE, the objective function is evaluated. This is done by performing a CFD simulation for each experiment in the DoE. Here the CFD Block is used to obtain the values of C_L, C_D, C_M which are used to evaluate the lift-constrained drag minimization problem. An array of function values is obtained which correspond to the responses obtained for each of the experiments.
4. RSM Construction—Based on the DoE and response values (steps 3 and 4), an over-determined problem is formulated $[\Phi][\beta] = [\mathbf{y}]$. This problem is then solved for $[\beta]$, and the RSM, $\hat{\mathbf{y}} \equiv \hat{f}(\mathbf{x}, \beta)$, is constructed.

5. RSM Optimization—Using the MATLAB Optimization and Global Optimization toolbox function, the RSM is optimized, and the optimum values of the design variables \mathbf{x}_{OPT} along with the corresponding objective function value J_{appr} (which is the function value approximated by the RSM) is obtained. Here the SQP algorithm is employed using MATLAB's `fmincon` and `MultiStart` functions. (`Mult_start_opt.m`)
6. Model verification—For the verification of the model, an additional CFD simulation is performed based on the values of \mathbf{x}_{OPT} . Here the airfoil geometry corresponding to the values of \mathbf{x}_{OPT} is obtained, and is sent to the CFD Block where the simulation is performed. The results obtained from this run are used to evaluate the actual objective function J_{actual} . This value is then compared with the approximated value J_{appr} .
7. If the difference between J_{appr} and J_{actual} is within a tolerance limit, then \mathbf{x}_{OPT} is the final optimum and the corresponding airfoil shape is the optimized design. Else, the iterative RSO methodology is implemented. Here \mathbf{x}_{OPT} and the corresponding J_{actual} is added to the initial DoE and the RSM is reconstructed and optimized. This process is repeated till the difference between J_{appr} and J_{actual} does not lie within the tolerance limit. Here the tolerance is set at 5 percent of J_{actual} .

The entire process is managed by a main directory called the MATLAB MASTER, which includes all the M-Files that are required to carry out the airfoil optimization process. Some of the important M-Files include—

- `bsp_basic.m` —Code for the generating a B-spline curve given the coordinates of the control points, parameter values and the order of the curve. This code returns the x and y coordinates of the B-spline curve.
- `optimum_fit.m`—Code for the parameterization of the baseline airfoil. Given the original coordinates of the airfoil geometry and a curve order, this code returns the coordinates of the control points and parameter values of the B-spline curve that best fits the baseline airfoil geometry. The y -ordinates of the control points and the angle of attack, the constitute the vector of design variables \mathbf{x} .

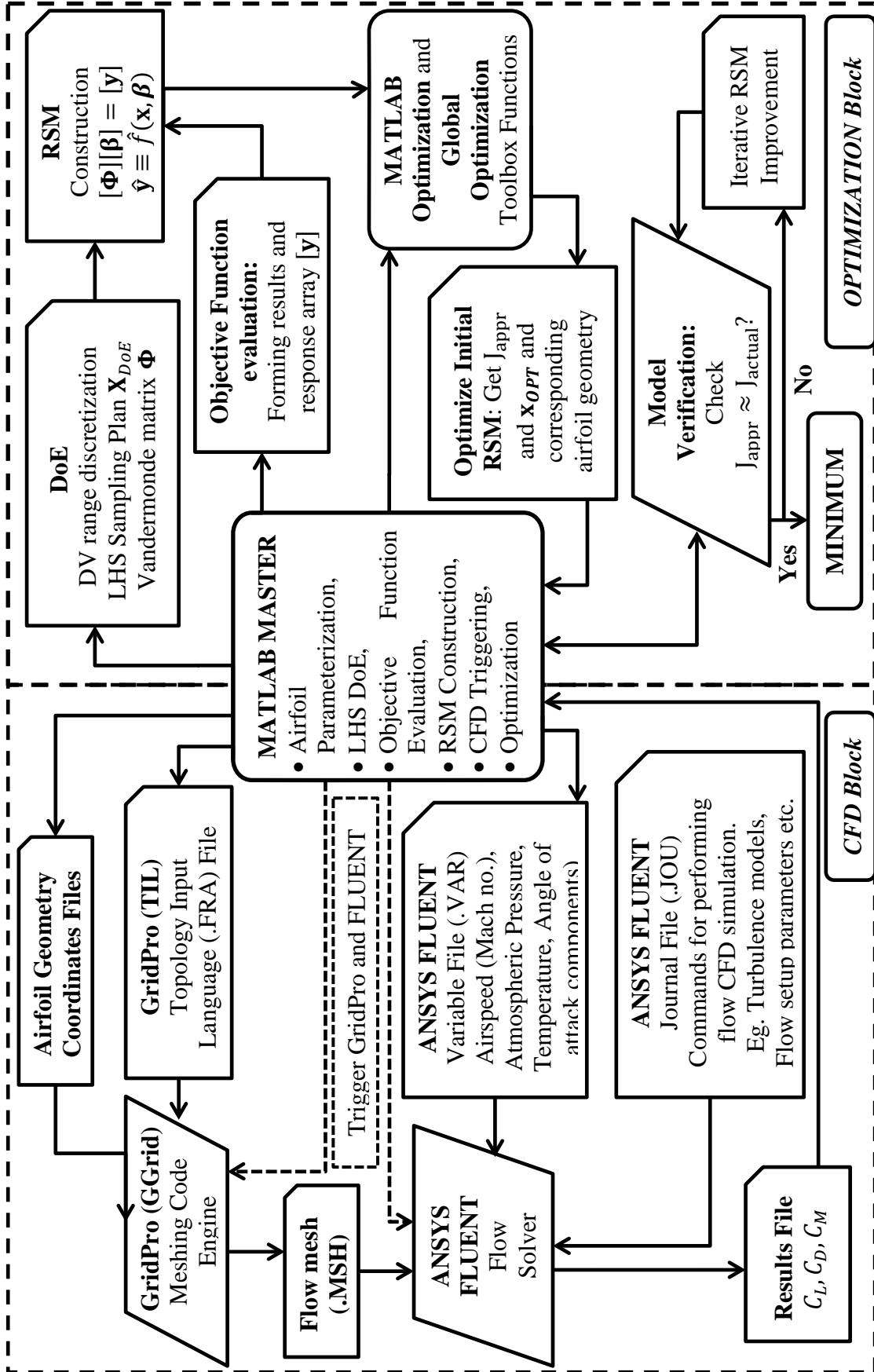


Figure 15 MATLAB implementation of the Iterative RSO method

- `CL_req.m`—Code for calculating the required coefficient of lift (C_L) given the weight, airspeed (Mach number) and altitude (in meters).
- `LH_doe.m`—Code for generating a LHS DoE plan required for constructing the RSM. Given the number of levels required, N_s , and the upper and lower limits of each of the design variables, the code returns a DoE set, \mathbf{X}_{DoE} consisting values of the design variables at all levels as per a LHS design.
- `Gramian.m`—Code for creating the Vandermonde matrix $[\Phi]$ and solving for β , given the values of the design variables at all levels \mathbf{X}_{DoE} and the corresponding responses $[\mathbf{y}]$.
- `J_approx.m`—Code for evaluating the RSM, $\hat{\mathbf{y}}$, given the values of β and design variables \mathbf{x} . This returns an approximate value of the objective function J_{appr} .
- `IterativeRSO.m`—Code for implementing the RSO based on the iterative methodology described above. Given \mathbf{X}_{DoE} , β and $[\mathbf{y}]$, the code performs the iterative RSO and returns the optimum values of the design variables \mathbf{x}_{OPT} .
- `NonlinCon.m`—Code for evaluating the non-linear constraints that are imposed on the RSM during optimization.
- `Multi_start_opt.m`—Code for performing global optimization using the Multi Start optimization algorithm. Here the optimization is performed from multiple starting points of the RSM leading to the optimum values.

3.4.1 MATLAB Optimization Toolbox

MATLAB Optimization Toolbox provides widely used algorithms for standard and large-scale optimization. These algorithms solve constrained and unconstrained continuous and discrete problems. The toolbox includes functions for linear programming, quadratic programming, binary integer programming, nonlinear optimization, nonlinear least squares, systems of nonlinear equations, and multiobjective optimization [39].

The function used here is *fmincon*. The purpose of this function is to find the minimum of a constrained nonlinear multivariable functions. It solves for a minimum of a problem specified by

$$\min_x f(x) \text{ such that } \begin{cases} c(x) \leq 0 \\ ceq(x) = 0 \\ A \cdot x \leq b \\ Aeq \cdot x = beq \\ lb \leq x \leq ub \end{cases}$$

Here x , b , beq , lb , and ub are vectors, A and Aeq are matrices, $c(x)$ and $ceq(x)$ are functions that return vectors, and $f(x)$ is a function that returns a scalar. $f(x)$, $c(x)$, and $ceq(x)$ can be nonlinear functions.

fmincon attempts to find a constrained minimum of a scalar function of several variables starting at an initial estimate. This is generally referred to as *constrained nonlinear optimization* or *nonlinear programming*.

The *fmincon* function syntax is as follows:

```
[x, fval] = fmincon(FUNC, x0, A, b, Aeq, beq, lb... ,ub,
NONLCON, OPTIONS)
```

This function call starts the optimization process from an initial guess value **x0** and attempts to find a minimizer **x** of the function described in **func** subject to the linear inequalities **A*x ≤ b**, equalities **Aeq*x ≤ beq**. It also defines a set of lower and upper bounds on the design variables in **x**, so that the solution is always in the range $lb \leq x \leq ub$. It subjects the minimization to the nonlinear inequalities $c(x)$ or equalities $ceq(x)$ defined in a function *NONLCON*.

Certain options required for the optimization such as the algorithm to be employed etc. is set using *OPTIONS*, which selects from a structure *OPTIMSET*. At the end of the optimization process, the function returns the value of the minimized **x** and the value of the corresponding objective value *fval*.

FUNC, is the objective function to be minimized. *FUNC* is a function that accepts a vector **x** and returns a scalar *f*, the objective function evaluated at **x**. *FUNC* can be specified as a function handle for a file:

```
[x] = fmincon(@myfunc, x0, A, b)
```

where *myfunc* is a MATLAB function such as

```
function f = myfunc(x)
```

```
f = ... % Compute function value at x
```

NONLCON is a function call that computes the nonlinear inequality constraints $c(x) \leq 0$ and the nonlinear equality constraints $ceq(x) = 0$. *NONLCON* accepts a vector \mathbf{x} and returns the two vectors \mathbf{c} and \mathbf{ceq} . \mathbf{c} is a vector that contains the nonlinear inequalities evaluated at \mathbf{x} , and \mathbf{ceq} is a vector that contains the nonlinear equalities evaluated at \mathbf{x} . *NONLCON* should be specified as a function handle to a file or to an anonymous function, such as `mycon`:

```
x = fmincon(@myfunc, x0, A, b, Aeq, beq, lb, ub, @mycon)
```

Here `mycon` is a MATLAB function such as

```
function [c, ceq] = mycon(x)
```

```
c = ... % Compute nonlinear inequalities at x.
```

```
ceq = ... % Compute nonlinear equalities at x.
```

`OPTIONS`, selects certain optimization parameters from a structure called `OPTIMSET`. Some of these options include—

- `Algorithm`—For selecting the optimization algorithm such as SQP etc.
- `DerivativeCheck`—For comparing the values of user-supplied derivatives to finite-differencing derivatives.
- `GradObj`—For using the gradient for the objective supplied by the user.
- `MaxFunEvals`—For specifying the maximum number of objective function evaluations.
- `MaxIter`—For specifying the maximum number of iterations allowed.
- `TolFun`—For specifying the termination tolerance on the function value.

The function call employed here is as follows:

```
f=@(x)rsm_opt(B, x);
```

```
nonlcon=@(x)NonlinCon(B,x,J);
```

```
[x_opt, fval]=fmincon(f, x0,[],[],[],[], lb, ub, nonlcon...  
,options);
```

Here `f` is a parameter that calls the function `rsm_opt(B, x)` which takes in values of the design variables \mathbf{x} and the regression coefficients $\boldsymbol{\beta}$, and evaluates the RSM equation. `nonlcon` is a parameter that calls the function

`NonlinCon(B,x,J)` that takes in values of \mathbf{x} , $\boldsymbol{\beta}$, and the actual value of the objective function at the previous step J , and evaluates the non-linear constraints functions.

The `fmincon` function call starts the optimization process at \mathbf{x}_0 (initial values of the design variables to initiate the optimization process), and performs the optimization iterations such that the iterate at each step lies within the bounds `lb` and `ub`, and satisfies the nonlinear constraints `nonlcon`. Following are the options that are used in the function call—

```
options=optimset('GradObj','on','GradConstr','on','algorithm','sqp','MaxFunEvals',10000000,'MaxIter',4000,'TolCon',... 0.00001,'TolX',0.00001);
```

The above options imply that the gradient of the objective function and the constraints are user-supplied, the algorithm to be used is SQP, maximum number of function evaluations are set at 10000000, maximum number of iterations are set at 4000, the constraints tolerance is set at 0.00001 and the tolerance for the change in the values of the design variables is set at 0.00001.

3.4.2 MATLAB Global Optimization Toolbox

Global Optimization Toolbox provides methods that search for global solutions to problems that contain multiple maxima or minima. It includes Global Search, Multistart, Pattern Search, Genetic Algorithm, and Simulated Annealing solvers [40].

These solvers can be used to solve optimization problems where the objective or constraint function is continuous, discontinuous, and stochastic, does not possess derivatives, or includes simulations or black-box functions with undefined values for some parameter settings.

Generally, Optimization Toolbox solvers find a local optimum. (This local optimum can be a global optimum). They find the optimum in the *basin of attraction* of the starting point. In contrast, Global Optimization Toolbox solvers are designed to search through more than one basin of attraction.

The GlobalSearch and MultiStart solvers apply to problems with smooth objective and constraint functions. The solvers search for a global minimum, or for a set of local minima. GlobalSearch and MultiStart work by starting a local solver, such as fmincon, from a variety of start points. In this case the MultiStart global optimization function is employed. Following are the functions used to perform optimization using the MultiStart function—

- CustomStartPointSet.m—This is a set of starting points (starting values of the design variables \mathbf{x}) that is supplied the MultiStart solver. By doing so, the MultiStart Solver starts a local optimization process from each of the supplied starting points. These points could be random points or user-supplied.
- createOptimProblem.m—This function call is used to setup the MultiStart problem. This is required in order to specify the local optimization solver to employed, objective functions, constraints, bounds, options etc.
- run.m—This function call runs the MultiStart solver based on the problem structure and the starting points defined above.

The functions used in this implementation are as follows:

```

tpoints = CustomStartPointSet([X_pnts]);
problem = createOptimProblem('fmincon','x0',x0_2,...
    'objective',@(x)rsm_opt(B, x ),'lb',lb,'ub',ub,...
    'nonlcon',@(x)NonlinCon(B,x,J),'options',options);
ms=MultiStart;
[xmin, J_min]=run(ms,problem,tpoints);

```

Here tpoints is the set consisting of the starting points defined by the CustomStartPointSet function, using the supplied points X_pnts.

problem is the optimization structure defined by the createOptimProblem function. It implies that the local optimization solver to be employed is the fmincon solver, the objective function is defined by the function rsm_opt(), the lower and upper bounds are defined by lb and ub, the nonlinear

constraints are defined by `nonlcon`, and the local minimization options are defined by `options`. Here `fmincon`, `rsm_opt()`, `nonlcon` are just as defined in the previous sub-section.

`ms` sets the global optimization solver to `MultiStart`. `run` performs the optimization process using `ms`, `problem` and `tpoints` that are defined above. At the end of the process, the lowest value of the objective function, `J_min` (which is the least amongst all the minima), and the corresponding values of the design variables `xmin` are returned.

3.5 Meshing using GridPro

For performing an external flow simulation using the CFD technique, the region around the airfoil to be meshed (spatial discretization) into quadratic cells. Here an automatic grid generator—GridPro is used to generate the airfoil meshes which are then used to carry out the flow simulation using ANSYS-Fluent.

For a multi-block structured grid generator, automation can be classified into four areas: 1). optimum distribution of high quality grid, 2). book keeping of topological information, 3). topology generation, and 4). surface restructuring and repair.

To this end, GridPro is a general purpose, 3-dimensional, multi-block structured grid (mesh) generator using an advanced smoothing scheme that incorporates many automatic features.

To generate a mesh, the input required from a user has three components—surface specifications, a block topology and a run schedule. These are described as follows:

3.5.1 Surface Specifications

The surface specifications constitute an external component in that they are provided mostly from the outside of the GridPro environment and conform to certain standards. The surface is basically the geometry around which the meshing is to be done and they are independent of the surface grid generated within GridPro. These surfaces are basically data files (file extension `.dat`) that contain the x , y and z coordinates surface points. The first line of the file displays the number of points used to describe the surface followed by the x , y and z coordinates of the points in three

columns. These files are then imported into the GridPro topology engine via the Topology Input Language (TIL) file.

For meshing the airfoil a C-type grid is to be generated. The C-type refers to the C-shape of the far-field boundary. The radius of this boundary is taken as 20 times the chord-length of the airfoil. The following surfaces were used to build grid—

- A C-shaped surface (dark blue) to define the boundary of the grid (filename: C_L.dat). This surface is assigned the pressure far-field property boundary condition so as to be recognized accordingly by the Flow-solver (ANSYS FLUENT).
- A rear-end surface (purple) so as to limit the boundary at the rear of the airfoil. This surface is also assigned the pressure far-field boundary condition property (filename: Back_end.dat).

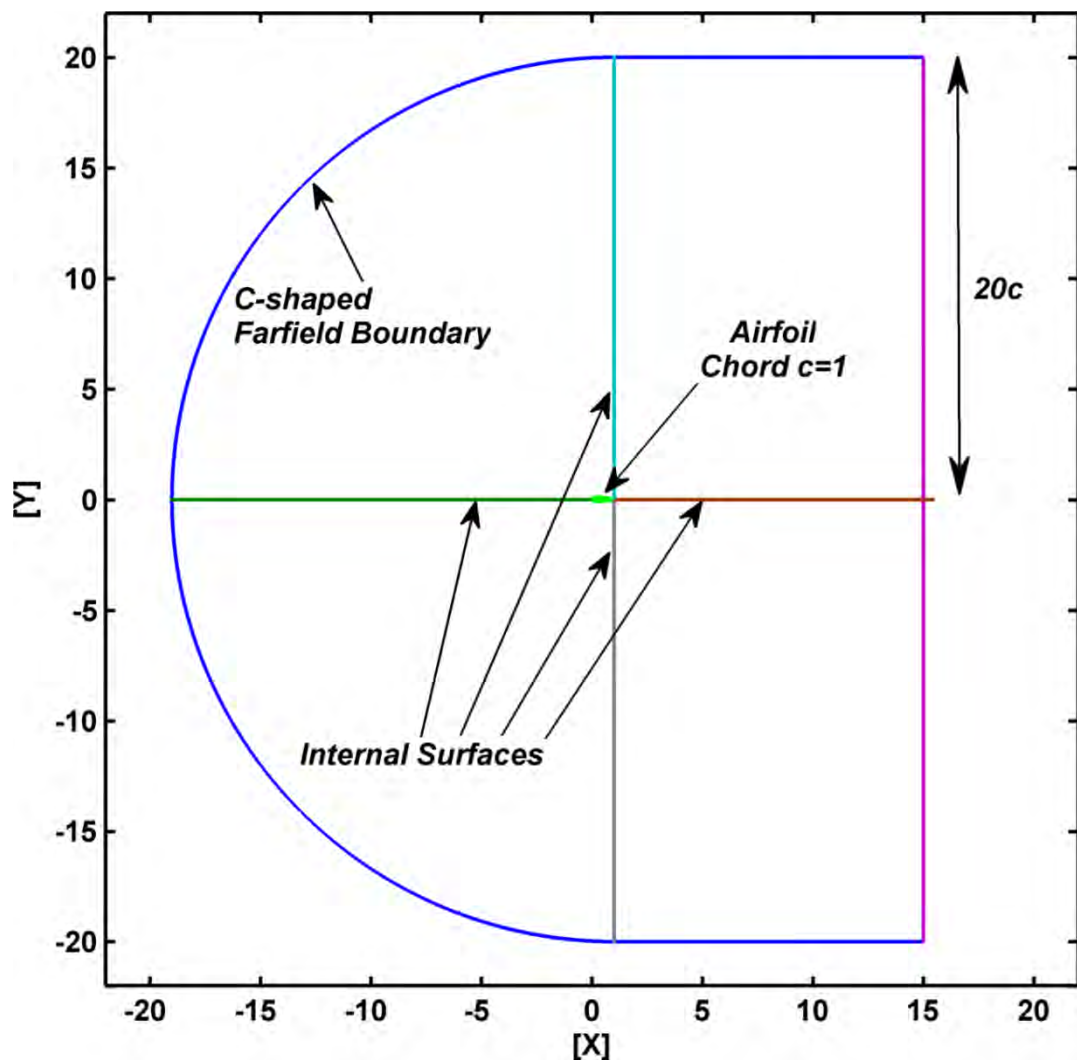


Figure 16 GridPro surfaces

- The airfoil surface (light green) is split into two—an upper surface and a lower surface, with a coordinates data file for each of them (GPro_af1.dat and GPro_af2.dat). These surfaces are assigned the wall property so that the flow solver applies the appropriate wall boundary conditions to them.
- In addition to the above surfaces, certain internal surfaces are specified. An internal surface is a surface for which both sides of the surface are to be gridded. These surfaces are used to provide the required grid point distribution. If a region within the grid is over stretched or if the geometry contains great disparity in length scales and curvatures, internal surfaces are used to separate the regions of disparity and get better grid distribution. Here, it is also used to capture the features of the airfoil wakeline (aft of the airfoil). This is done by creating a surface which starts from the airfoil trailing edge, and extends through to the rear end surface. Also, this surface must be created such that it bisects the airfoil trailing edge angle. Using this internal surface also ensures a sharp trailing edge for the airfoil (filenames: af_lead.dat, trail_top.dat, trail_down.dat, Wake_line.dat).

3.5.2 Block Topology

After the surfaces have been defined, the next task is to design a block topology for the region to be gridded. This can be done with the GridPro/az-Graphic Manager, which is the graphical user interface for the GridPro engine.

The goal is to cover the region with quadrilaterals (blocks). This covering does not need to be done at a geometric precise level. It needs to be done only at a rather topological level. That is, a surface can be represented as a set of piece-wise linear segments placed not too far from the real surface.

It can be noticed that the entire region between the outer far-field and the airfoil at the center has been filled with blocks (quadrilaterals). Here the points in orange are referred to as corners and the yellow line segments as block edges. Each corner in the interior of the domain is shared by four neighboring blocks and each block edge is shared by two blocks. The corners and edges are arranged such that quadrilateral that are eventually formed capture the topological features (geometry) of the surfaces.

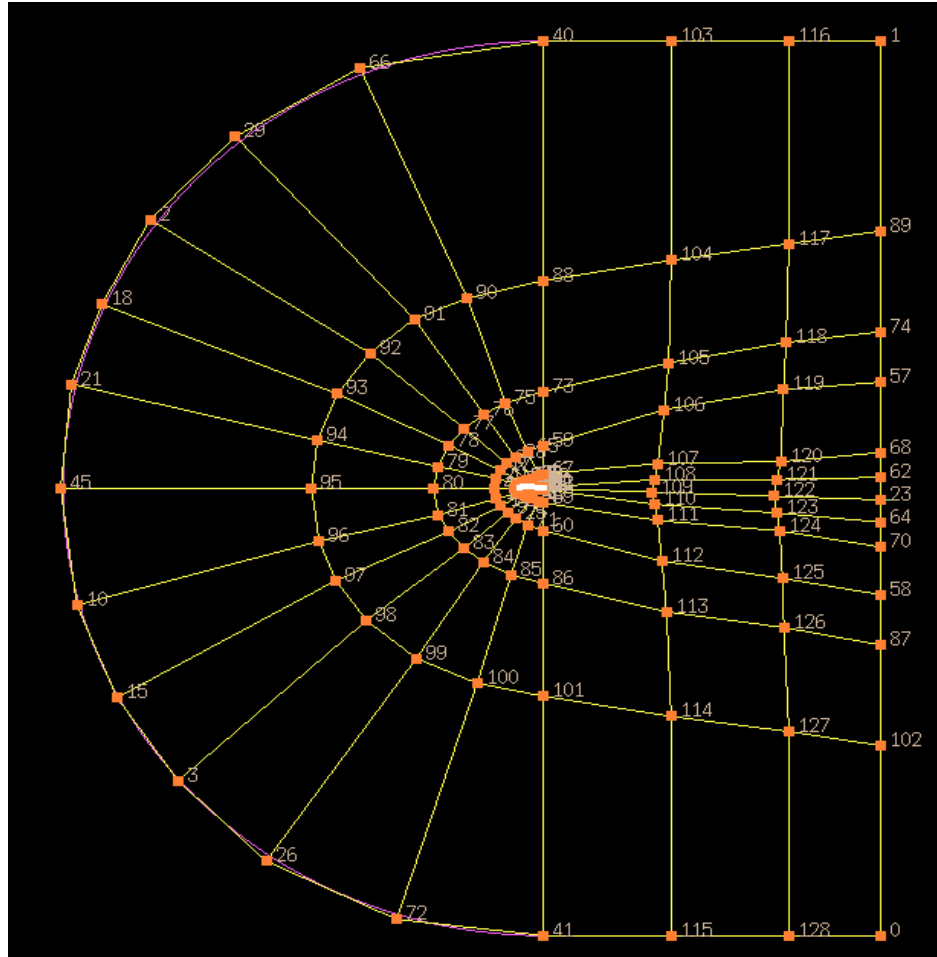


Figure 17 Topology layout for the airfoil and the far-field

The topology building is started from the outer far-field by placing the corners (numbered 0, 128, 115 and so on) around in close proximity to the outer surfaces. These corners are then linked to form edges. Corners (numbered 127, 114, 101 and so on) are added in the interior of the domain in the manner shown in the above figure. These corners are then connected to each other and to the corners surrounding the outer surfaces such that blocks (quadrilaterals) are formed. More corners and edges are placed in the interior of the domain closing in on to the center such that blocks now begin to wrap around the airfoil surface and the wake-line.

The region surrounding the airfoil and the wake-line is where the flow is highly dynamic and turbulent. So it is important to capture the geometrical features in these regions with greater accuracy and thus more blocks are used to wrap the airfoil. Also higher grid densities (number of cells emerging from the edge) are assigned to edges that make up these blocks. This leads to a relatively denser grid near the airfoil surface and the wake line.

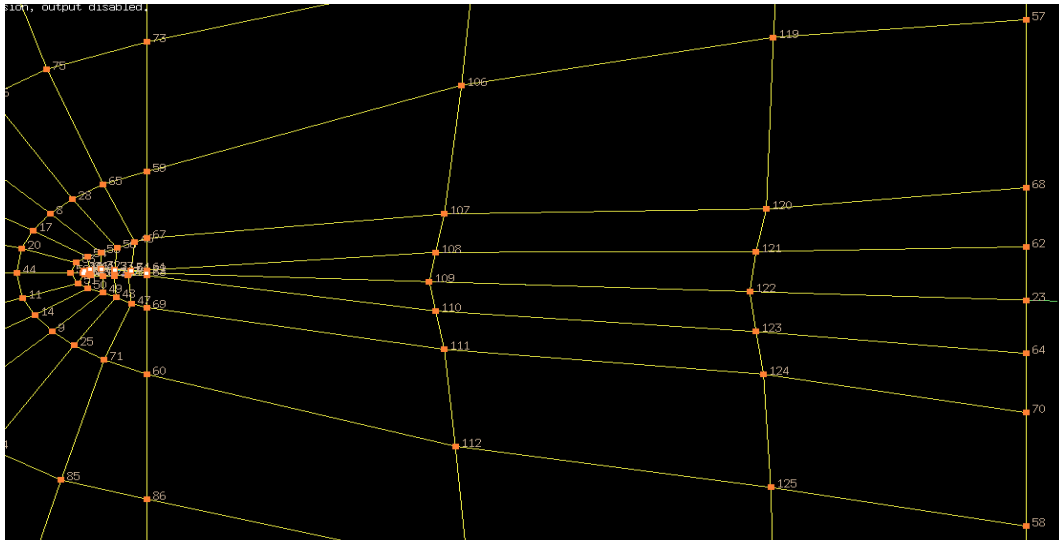


Figure 18 Topology layout in the wake region of the airfoil

It is desirable to have the blocks as orthogonal as possible in the regions surrounding the airfoil so that the resulting grid around it is also orthogonal. This leads to more accurate CFD results. In order to do so, a MATLAB code is written which adjusts the position of the corners surrounding the airfoil and the wake-line. The corners immediately surrounding the airfoil are positioned such that the edges emerging from them (radially) are perpendicular to the airfoil surface. Also, the arrangement of these corners ensures that the edges connecting them are approximately tangential to the surface. This way, the blocks are almost tangential to the airfoil surface and thus the cells that are formed are almost orthogonal to the surface.

Once the topology design and layout is complete, the corners in the immediate vicinity of the surface are assigned to it. For instance, the corners that are closest to the airfoil surface are assigned to it. This way the GridPro grid engine recognizes the points which are associated to surfaces and wraps the grid around them accordingly.

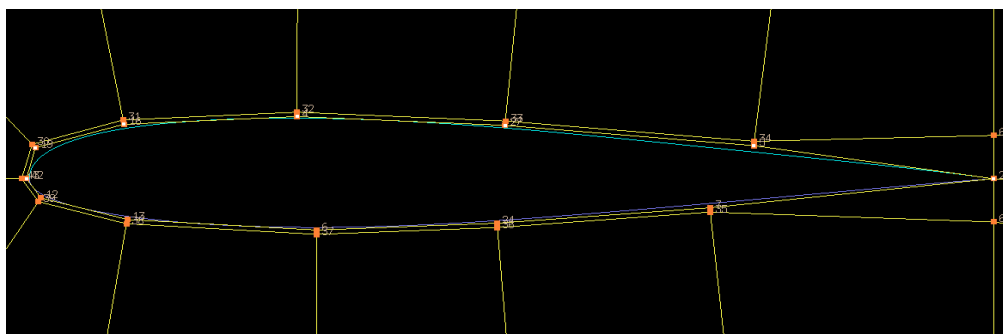


Figure 19 Topology around the airfoil surface

After the completion of the topology creation and the assignment of the respective surface properties, the topology is saved as a Topology Input Language (TIL) file (extension .FRA). This file keeps a record of the coordinates of all the corners, edges, surface filenames, properties, grid densities etc. and is used to run the gridding process. (Filename: FINAL3.FRA).

3.5.3 Run schedule

Since GridPro uses an advanced smoothing scheme in which the grid is generated in multiple sweeps just as in the case of the ordinary elliptic grid generation, a schedule is provided for a better and faster convergence. Here 1000 sweeps are used to generate the grid.

A schedule file consists of step lines, each of which lists a sequence of actions that direct the run process of GridPro. The name of a schedule file has the prefix part same as the corresponding main topology file and end with the file name extension '.sch' (Filename: FINAL3.sch).

The GGrid engine is run which executes the TIL file and performs the specified number of sweeps. At the end of the sweeping process a temporary block file is generated (extension .tmp). This file includes the coordinates of the grid points that are generated at the last sweep.

3.5.4 Wall Clustering

After the grid is generated (after executing the TIL file with the sweeps), wall clustering is used to adjust the spacing of the cells near the airfoil surface such that there is a much denser grid near the surface. This is particularly important in order to capture the flow physics associated with turbulent and separated flows.

Wall clustering reduces the normal spacing of the cells attached to the wall surfaces to a much lower value. It then starts increasing the spacing of the cells in the layers adjacent to the wall layer by a stretch factor called the growth ratio. This way there is a gradual increment in the cell spacing starting from the wall surface going towards the outer far field. This is done using the `-clu` command. Here the spacing of the wall cells (cells attached to the wall surface) is set at 0.00025 and the growth ratio is set at 1.05.

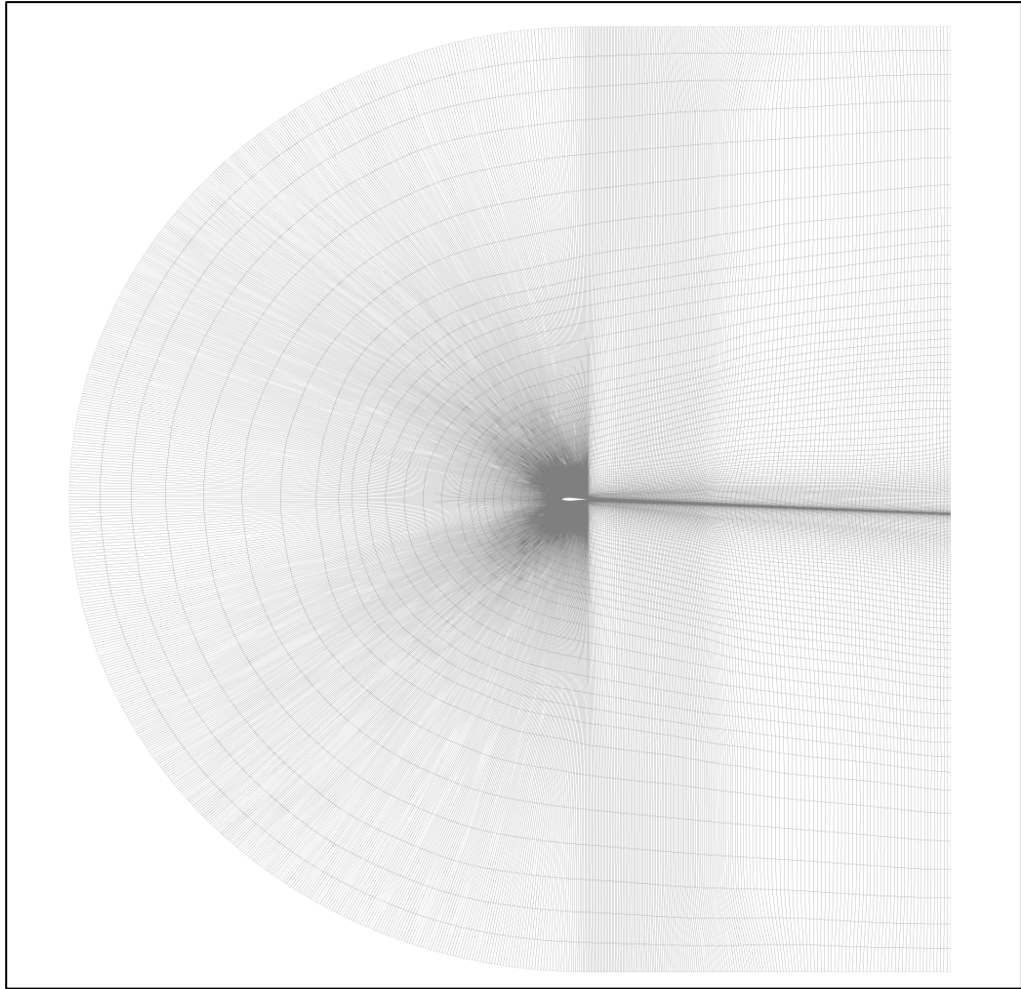


Figure 20 Mesh—full display

After the clustering is performed, the temporary grid is now saved in the required format which can be read by the flow solver. The ANSYS FLUENT has a specific format for the grid which is given by the (.MSH) extension. A FLUENT output script (outU_fluent.script.bat) is run inside GridPro which converts the temporary (.tmp) grid file into the required (.MSH) format. This mesh can then be imported into ANSYS FLUENT to perform the required simulation.

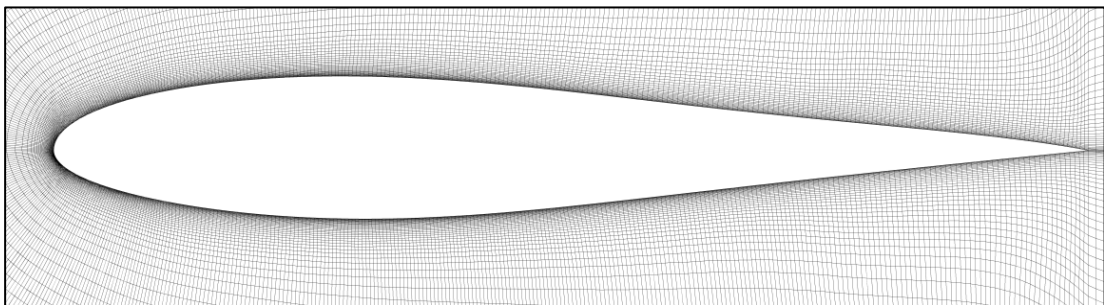


Figure 21 Mesh—full and close up view

3.6 CFD Solver—ANSYS FLUENT

ANSYS FLUENT is a flow solver that provides comprehensive modeling capabilities for a wide range of incompressible and compressible, laminar and turbulent fluid flow problems. Steady-state or transient analyses can be performed. In ANSYS FLUENT, a broad range of mathematical models for transport phenomena (like heat transfer and chemical reactions) is combined with ability to model complex geometries [31].

Robust and accurate turbulence models are a vital component of the ANSYS FLUENT suite of models. The turbulence models provided have a broad range of applicability, and they include the effects of other physical phenomena, such as buoyancy and compressibility.

For all flows, ANSYS FLUENT solves conservation equations for mass and momentum which are the basic governing equations for any flow. For flows involving heat transfer or compressibility, an additional equation for energy conservation is solved. In addition to these, turbulence model equations are solved to account for the turbulence associated with aerodynamic flows. The turbulence model used here is the Spalart-Allmaras model. It is a one-equation turbulence model and it is solved for the turbulent kinematic viscosity $\tilde{\nu}$, which can then be used to calculate the turbulent dynamic viscosity, μ_t , that will eventually close the RANS equations. All these equations have been discussed in depth in Chapter 2.

In ANSYS FLUENT, the flow can be solved using one of the two numerical methods— Pressure-based solver and the Density-based solver. In either of the cases a control-volume based technique is used that consists of—

- Division of the domain into discrete control volumes using a computational grid.
- Integration of the governing equations on the individual control volumes to construct algebraic equations for the discrete dependent variables (unknowns) such as velocities, pressure, temperature, and conserved scalars.
- Linearization of the discretized equations and solutions of the resultant linear equation system to yield updated values of the dependent variables.

3.6.1 Pressure-Based Solver

In the pressure-based approach, the pressure field is extracted by solving a pressure or pressure correction equation which is obtained by manipulating the continuity and momentum equations. Each iteration of the solver consists of the following steps

- Update fluid properties (eg. density, viscosity, turbulent viscosity etc.)
- Solve simultaneously the coupled system of equations—momentum and pressure based continuity equations.
- Correct face mass fluxes, pressure and the velocity field.
- Solve the equations for additional scalars, such as turbulent quantities etc,
- Check for convergence of the equations.

These steps are continued until the convergence criteria are met.

3.6.2 General Scalar Transport Equation

All the governing differential equations for fluid motion, which are based on the conservation principles (conservation of mass, momentum and energy) can be written in a generic form. Upon inspection of the all the conservation equations, it can be inferred that all the dependent variables seem to obey a generalized conservation principle. If the dependent variable (scalar) is denoted by ϕ , the generic differential equation is given as

$$\underbrace{\frac{\partial \rho \phi}{\partial t}}_{\text{Transient term}} + \underbrace{\nabla \cdot (\rho \mathbf{V} \phi)}_{\text{Convection Term}} = \underbrace{\nabla \cdot (\Gamma \nabla \phi)}_{\text{Diffusion term}} + \underbrace{S_\phi}_{\text{Source term}}$$

Here Γ is the diffusion coefficient or diffusivity.

- The transient term, $\frac{\partial \rho \phi}{\partial t}$, accounts for the accumulation of ϕ in the concerned control volume.
- The convection term, $\nabla \cdot (\rho \mathbf{V} \phi)$, accounts for the transport of ϕ due to the existence of the velocity field.
- The diffusion term, $\nabla \cdot (\Gamma \nabla \phi)$, accounts for the transport of ϕ due to its gradients.
- The source term, S_ϕ , accounts for any sources or sinks that either create or destroy ϕ . Any extra term that cannot be cast into the convection or

diffusion terms are considered as source terms. For chemically inert flows, such as aerodynamic flows, where ideal gases are assumed in the calculations, the S_ϕ term is equated to zero.

The objective of all discretization techniques (here the finite volume method) is to devise a mathematical formulation to transform each of these terms into an algebraic equation. Once applied to all control volumes in a given mesh, a full linear system of equations is obtained that needs to be solved for the variable ϕ .

ANSYS FLUENT uses a control-volume based technique to convert a general scalar transport equation to an algebraic equation can be solved numerically. This control volume technique consists of integrating the transport equation about each control volume, yielding a discrete equation that expresses the conservation law on a control-volume basis.

This is demonstrated by the following equation (which is the general transport equation) written in integral form for an arbitrary control volume V as follows:

$$\int_V \frac{\partial \rho \phi}{\partial t} dV + \oint \rho \phi \mathbf{V} \cdot \mathbf{dA} = \oint \Gamma_\phi \nabla \phi \cdot \mathbf{dA} + \int_V S_\phi dV$$

Where ρ = density

\mathbf{V} = velocity vector ($\mathbf{V} = u\mathbf{i} = v\mathbf{j}$)

\mathbf{dA} = differential surface area vector

Γ_ϕ = diffusion coefficient for ϕ

$\nabla \phi$ = gradient of $\phi = (\partial \phi / \partial x)\mathbf{i} + (\partial \phi / \partial y)\mathbf{j}$

S_ϕ = source for ϕ per unit volume

The above equation is applied to each control volume, or cell, in the computational domain. Discretization of the above equation on a given cell yields

$$\frac{\partial \rho \phi}{\partial t} V + \sum_f^{N_{faces}} \rho_f \mathbf{V}_f \phi_f \cdot \mathbf{A}_f = \sum_f^{N_{faces}} \Gamma_\phi \nabla \phi_f \cdot \mathbf{A}_f + S_\phi V$$

Here

- N_{faces} = number of faces enclosing cell (for a quad cell $N_{faces} = 4$).

- ϕ_f = value of ϕ convected through face f .
- $\sum_f^{N_{faces}} \rho_f \mathbf{V}_f \phi_f \cdot \mathbf{A}_f$ = mass flux through the face.
- \mathbf{A}_f = area vector of the face ($\mathbf{A}_f = A_x \mathbf{i} + A_y \mathbf{j}$)
- $\nabla \phi_f$ = gradient of ϕ at face f .
- V = cell volume.

The discretized scalar transport equation contains the unknowns scalar variable ϕ at the cell center as well as the unknown values in surrounding neighbor cells. This equation will, in general, be non-linear with respect to these variables. A linearized form can be written as

$$a_p \phi = \sum_{nb} a_{nb} \phi_{nb} + b$$

Here the subscript nb refers to neighbor cells, and a_p and a_{nb} are the linearized coefficients for ϕ and ϕ_{nb} . The number of neighbors for each cell depends on the mesh topology, but will typically equal the number of faces enclosing the cell (with the exception of the boundary cells).

Similar equations can be written for each cell in the mesh. This results in a set of algebraic equations with a sparse co-efficient matrix. For scalar equations, ANSYS FLUENT solves the linear system using a point implicit (Gauss-Siedel) linear equation solver in conjunction with an algebraic multigrid (AMG) method.

3.6.3 Spatial Discretization

By default, ANSYS FLUENT stores discrete values of the scalar ϕ at the cell centers. However, face values ϕ_f are required for the convection terms and must be interpolated from the cell center values. This is accomplished using an upwind scheme.

Upwinding means that the face value ϕ_f is derived from quantities in the cell upstream, relative to the direction of the normal velocity \mathbf{V}_f . There are several upwind schemes that are available—first-order upwind, second-order upwind, power law and QUICK. Here the second-order upwind scheme is used.

In this scheme, the quantities at the cell faces are computed using a multidimensional reconstruction approach. In this approach, higher-order accuracy is achieved at cell faces through a Taylor series expansion of the cell-centered solution about the centroid. Thus the face value ϕ_f is computed using the following expression—

$$\phi_f = \phi + \nabla\phi \cdot \mathbf{r}$$

Here ϕ and $\nabla\phi$ are the cell-centered value and its gradient in the upstream cell, and \mathbf{r} is the displacement vector from the upstream cell centroid to the face centroid. This formulation required the determination of the gradient $\nabla\phi$ in each cell.

3.6.4 Computing Forces and Moments

For the wall surfaces, the forces along a specified vector and the moments about a specified center and axis are computed.

The total force component along a specified force vector \mathbf{a} on a wall surface (airfoil surface) is computed by summing the dot product of the pressure and viscous forces on each surface with the specified force vector. The terms in this summation represent the pressure and viscous force component in the direction \mathbf{a} :

$$\underbrace{F_a}_{\text{Total force component}} = \underbrace{\mathbf{a} \cdot \mathbf{F}_p}_{\text{pressure force component}} + \underbrace{\mathbf{a} \cdot \mathbf{F}_v}_{\text{viscous force component}}$$

Here \mathbf{a} = specified force vector

\mathbf{F}_p = pressure force vector

\mathbf{F}_v = viscous force vector

The total moment vector about a specified center A is computed by summing the cross products of the pressure and viscous force vectors for each surface with the moment vector \mathbf{r}_{AB} , which is the vector from the specified moment center A to the force origin B . The terms in this summation represent the pressure and viscous moment vectors.

$$\underbrace{\mathbf{M}_A}_{\text{Total moment}} = \underbrace{\mathbf{r}_{AB} \times \mathbf{F}_p}_{\text{pressure moment}} + \underbrace{\mathbf{r}_{AB} \times \mathbf{F}_v}_{\text{viscous moment}}$$

Here

A = specified moment center

B = force origin

\mathbf{r}_{AB} = moment vector

In addition to the forces and the moments, the associated force and moment coefficients are also computed for each of the wall surfaces, using the reference values.

The force coefficient is defined as the force F_a divided by $\frac{1}{2}\rho v^2 A$, and the moment coefficient is defined as magnitude of the moment M_A divided by $\frac{1}{2}\rho v^2 AL$. Here ρ , v , A , and L are the density, velocity, area the length of the surface.

For calculating the lift and drag on the airfoil, the vector \mathbf{a} is the unit vector perpendicular and along the direction of the airflow respectively. For calculating the moment, \mathbf{r}_{AB} is the distance vector between the line of action of force and the reference point, taken to be $0.25L$.

3.6.5 ANSYS FLUENT Setup and Run

To perform each CFD run, certain steps need to be taken. These are as follows—

1. Import the mesh file into the solver—Here the mesh file that is generated using GridPro is imported into the flow solver using the Read Mesh command.
2. Define viscous model—Here the turbulence model (Spalart-Allmaras) with the default values is selected.
3. Define fluid material—Here the type of fluid properties are selected (ideal gas, air). For defining the viscosity, the Sutherland (3-Co-efficient) method is used. The ideal gas property automatically activates the Energy Equation option.
4. Merge-zones—Certain zones in the mesh are merged into one, so that the total number of zones are reduced. In particular, the pressure far-field and the interior zones are merged.
5. Load variable file—The variable file is loaded into the solver. This contains variable values for the operating pressure (P), Temperature (T), Mach number (M), unit vector magnitudes for the x and y components of the airflow (M_x and M_y), unit vector magnitudes for the x and y

components of the Lift force (L_x and L_y), and unit vector magnitudes for the x and y components for the Drag force (D_x and D_y).

6. Define boundary conditions—Here the boundary conditions are defined using the variables that are loaded into the solver (step 5). Here Mach number and its x and y components, and temperature are defined by assigning the corresponding variables to the respective fields. Also the turbulence specification method is set to Turbulent-Viscosity ratio and the value is set at 10. The operating pressure is defined by assigning the corresponding variable.
7. Set Solution methods—here the pressure-velocity coupling is set to coupled. For the Spatial Discretization, the gradient is computed using Least-Squares Cell Based scheme, Pressure is solved using the standard scheme, and density, momentum, modified turbulent viscosity and energy are solved using the second-order upwind scheme.
8. Set solution controls—The Courant Number is set at 200, the explicit relaxation factors for momentum and pressure are set at 0.5, and under-relaxation factor for density is set at 0.5, for body-forces, turbulent-viscosity and energy it is set at 1, and for modified turbulent viscosity it is set at 0.9.
9. Set monitors—Here the monitors for the continuity, x-velocity, y-velocity, energy and kinematic viscosity are set. The absolute convergence criteria for these variables are respectively set at $1e-07$, $1e-05$, $1e-05$, $1e-06$ and $1e-05$. In addition to these monitors, the lift, drag and moment residuals are also monitored.
10. Initialize—The flow field must be initialized in order to carry out the iterations. This is done by using the hybrid initialization option.
11. Solve—Set the number of iterations and run the solver. Here the number of iterations is set at 300.
12. Report forces—Once the solution has converged, the wall forces are reported and saved. The wall forces are the lift, drag and moments. For the calculation of the correct lift and drag values, the horizontal and vertical component unit vector magnitudes are specified by the variables L_x and L_y , D_x and D_y respectively. These variables take their values from the variable file which is loaded into the solver at the beginning of the

program. The moments are calculated about an axis perpendicular X-Y plane and at 25% of the chord length (LIFT.csv, DRAG.csv, and MOMENT.csv).

13. The pressure coefficient (CP), Mach number and shear distributions over the top and bottom surfaces of the airfoil are plotted as a function of distance along the airfoil chord. These distributions are also saved in individual files (CP_TOP.csv, CP_LOW.csv, Mach_TOP.csv, Mach_LOW.csv, Stress_TOP.csv, Stress_LOW.csv).

14. After saving the results, the solver is exited.

The steps mentioned above are recorded in a script file, which is referred to as a Journal file (extension .JOU). This file is read by ANSYS-FLUENT solver and the steps are executed automatically without requiring any input from the user. In this way complete automation is achieved which is the backbone of the parametric optimization process.

3.6.6 Flow Solver Validation

Flow results verification studies are performed in order to verify the results obtained from the flow solver. Here the results obtained from ANSYS-FLUENT are compared with those obtained by verified sources that are usually used by the CFD community. These resources have been published by NASA's National Program for Applications-Oriented Research in CFD (NPARC). NPARC provides an archive of examples, verification and validation cases for general purpose CFD usages.

The case used for verification here is a RAE 2822 airfoil at transonic conditions. This case is Study#4 from the NPARC Validation Archive. The freestream conditions for this case are:

- Mach Number = 0.729 [-]
- Static Pressure = 15.8073 [psi] = 108987.768 [N/m²]
- Temperature = 460.0 [R] = 255.55 [K]
- Angle of attack = 2.31°

A CFD mesh is obtained for the RAE 2822 airfoil using GridPro. The process is as described in the previous section. The mesh obtained is analyzed for the flow based on the steps described above and using the freestream conditions for this case-

study. The results obtained are compared with three results published by NPARC. These are—the NPARC solver code, WIND-US 3.0 code and published wind-tunnel

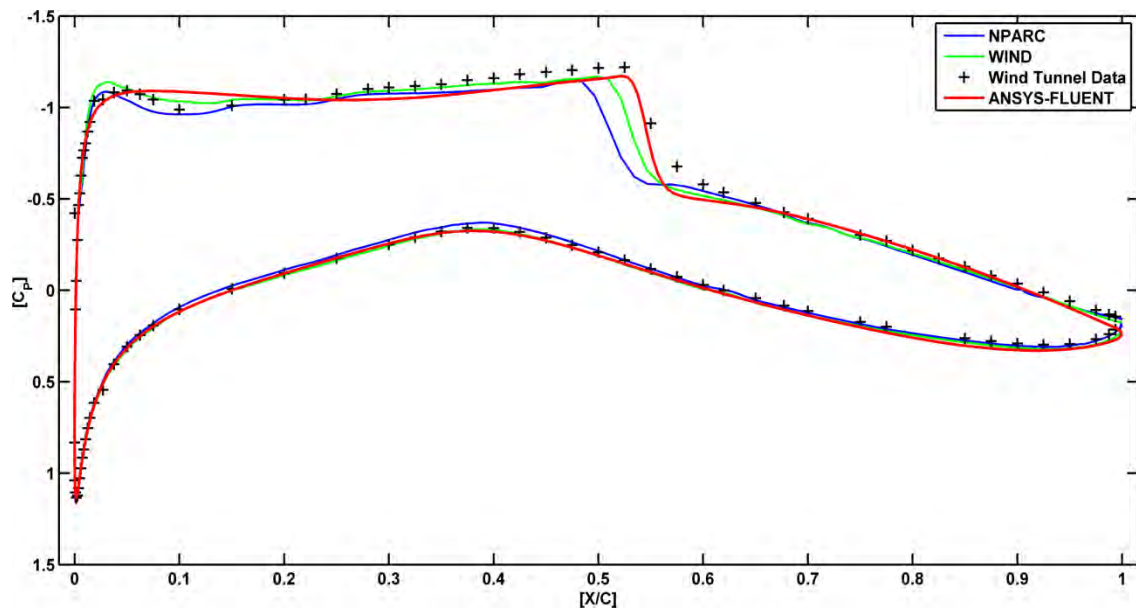


Figure 22 CFD verification results

data. Here, the pressure coefficient (C_p) distribution over the upper and lower surface obtained by FLUENT are compared with the published results as below:

It can be seen from the above plot that the results obtained by ANSYS-FLUENT are in close agreement with the published results.

3.7 Simulation Time

The processor used to perform the CFD simulations using ANSYS-FLUENT and GridPro, and the optimization processes using MATLAB has the following configuration—

- Intel® Core™ i5 CPU, M430
- Processor speed— 2.27 GHz
- Installed memory (RAM)— 8.00GB of which 7.86 GB is usable.
- System type— 64-bit operating system,
- Manufacturer— DELL

The time required for a generating a single mesh is 81.0847 seconds computer time, while the time required for performing a single CFD flow simulation on the mesh is 258.037225 seconds computer time. Therefore one complete CFD simulation (meshing and flow solving) takes 339.1219 seconds of computer time.

The overall meshing and simulation time for 225 samples is 76302.44973 seconds or 21.1951 hours computer time.

4. RESULTS AND DISCUSSIONS

4.1 Overview

The iterative RSO methodology is implemented here on various case studies. First a design validation study is carried out by implementing the scheme for the optimization of the RAE 2822 airfoil for a particular operating condition. Following this study, an adaptive airfoil is designed by optimizing a Boeing 737 airfoil for four different operating conditions. These operating conditions correspond to three pairs of cruising speeds and altitudes.

The objective function used here is the lift-moment constrained drag minimization problem. 11 design variables are used—the 10 B-spline control points, controlling the airfoil shape, and the angle of attack. The initial RSM is constructed using full quadratic polynomial regression comprising 78 terms and using 200 sampling points based on a LHS design. SQP optimization algorithm is used to optimize the RSM and the iterative RSO methodology is employed to improve the predictability of the RSM.

4.2 Case Study#1: Optimization of the RAE 2822 Airfoil

This study is used as a design validation study to compare the results obtained by the implementation of the iterative RSO methodology with results that have been published in the literature.

RAE 2822 is a transonic airfoil and it has extensively been used in the literature as a design validation platform for validating various optimization methodologies. The design condition of this airfoil is as follows—

- $M_\infty = 0.73[-]$
- $\alpha = 2.7^\circ$
- $Re = 7.196 \times 10^6[-]$
- Static pressure = 43765 [N/m²]
- Temperature = 300 [K]

For this condition the ANSYS-FLUENT solver has predicted a C_L value of 0.78902, a C_D value of 0.0166, and a C_M value of 0.0918. The viscous, Garabedian and Korn (VGK) flow-solver code employed by Armando et al. for the same conditions as above predicted a C_L value of 0.7894 and a C_D value of 0.0150 [38]. The

Navier-Stokes solver with Baldwin-Lomax turbulence model employed by Ahn et al. predicted the same value of the lift coefficient ($C_L=0.7894$) and a C_D value of 0.0193 at $\alpha = 2.7^\circ$ [34]. Despite the difference in the drag prediction, all three flow solvers are in good agreement as far as the pressure coefficient distribution is concerned.

The parameterization of the airfoil, as discussed earlier was carried out using B-spline control points. The y-ordinates of the control points 2 through 6 and 9 through 13 form the vector of design variables. In addition the angle of attack is taken as the eleventh design variable.

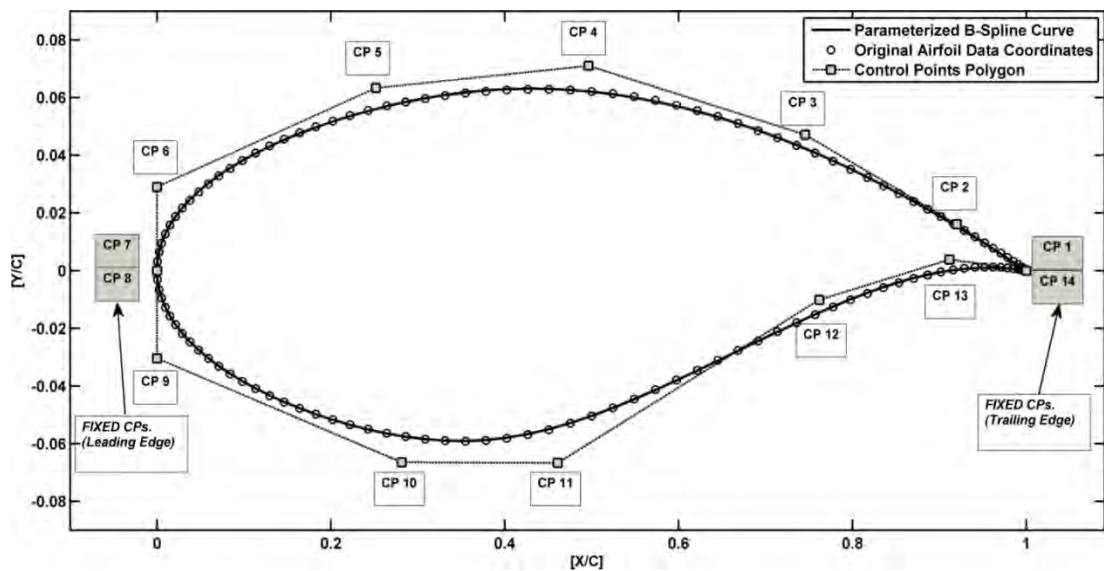


Figure 23 B-spline parameterization of the RAE 2822 airfoil

The range for each DV was obtained by defining an upper and a lower limit. These were taken as 50% above and below the baseline values respectively.

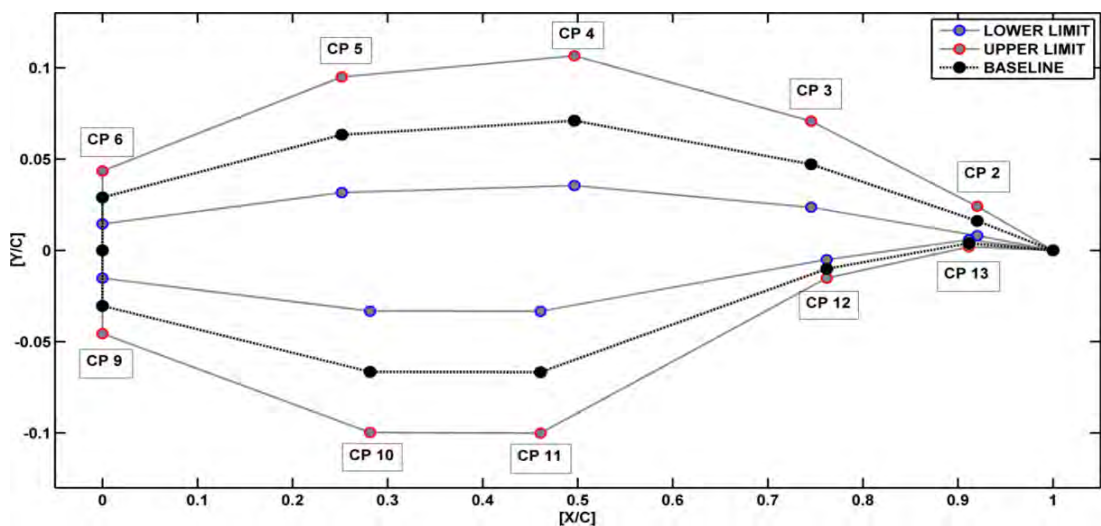


Figure 24 Range of the B-spline control points (Design variables)

The baseline, and the upper and lower limit values of the design variables are as follows:

Table 2 Baseline and Upper and Lower limit Values of the Design Variables

DV	CP	Baseline Values	Upper Limit	Lower Limit
1	2	0.0161118307	0.02416774612	0.0080559153
2	3	0.0471756500	0.07076347511	0.0235878250
3	4	0.0710297412	0.10654461194	0.0355148706
4	5	0.0633427087	0.09501406311	0.0316713543
5	6	0.0290058511	0.04350877666	0.0145029255
6	9	-0.0304069342	-0.04561040142	-0.0152034671
7	10	-0.0665061285	-0.09975919278	-0.0332530642
8	11	-0.0667191889	-0.10007878335	-0.0333595944
9	12	-0.0101262400	-0.01518936007	-0.0050631200
10	13	0.0038518768	0.00192593840	0.0057778152
11	Angle of Attack	2.70	2	3.4

Following are the freestream conditions for the RAE 2822 design conditions—

Table 3 Freestream conditions for the design operating condition

True Airspeed	Mach 0.73 [-]
	253.374 [m/s]
Atmospheric Pressure	43765 [N/m ²]
Air Density	0.508237 [kg/m ³]
Ambient Temperature	300 [K] (17 [°C])
Dynamic Viscosity	1.7894e-05 [kg/m-s]
Reynolds Number	7.2x10 ⁶ [-]
Angle of Attack	2.7°

The following values compare the performance parameters (Lift, moment and drag) of the baseline and optimized airfoil. Here (C-W) refers to Clockwise moment and (C-CW) refers to counter-clockwise moments—

Table 4 Baseline and Optimized performance at off-design operating point

PERFORMANCE PARAMETERS	BASELINE	OPTIMIZED	REMARKS
LIFT FORCE			
Pressure Lift [N]	12872.11	13254.782	
Viscous Lift [N]	-0.07309	-0.157937	
Total Lift [N]	12872.036 [REQUIRED]	13254.624	DIFFERENCE OF 2.97%(HIGHER)
Pressure CL [-]	0.78902	0.812478	
Viscous CL [-]	-4.4803e-06	-9.681e-06	
Total CL [-]	0.789017 [REQUIRED]	0.812469	DIFFERENCE OF 2.97%(HIGHER)
MOMENTS			
Pressure Moment [N-m]	1499.3075	1675.9315	
Viscous Moment [N-m]	-1.1438	-2.05919	
Total Moment [N-m]	1498.163 [C-CW] [REQUIRED]	1673.8723	DIFFERENCE OF 11.72%(LESS C- CW)
Pressure CM [-]	-0.091903	0.102729	
Viscous CM [-]	-7.011e-05	-0.00012622	
Total CM [-]	0.091832 [C-CW] [REQUIRED]	0.1026034	DIFFERENCE OF 11.72%(LESS C- CW)
DRAG FORCE			
Pressure Drag [N]	180.6682	113.355	
Viscous Drag [N]	90.24008	93.63396	
Total Drag [N]	270.9082	206.989	REDUCTION OF 23.59%
Pressure CD [-]	0.0110744	0.00694832	
Viscous CD [-]	0.0055314	0.0057394	
Total CD [-]	0.01660587	0.01268	REDUCTION OF 23.59%
Δ CD = 0.003925 \approx 4 [DRAG COUNTS] LESS			
ANGLE OF ATTACK	2.7°	2.24°	

Table 5 Baseline and Optimized values of the Design Variables

DV	CP	Baseline Values	Optimum Values
1	2	0.0161118307	0.0241677461253
2	3	0.0471756500	0.0425305732457
3	4	0.0710297412	0.0805926624195
4	5	0.0633427087	0.0525445332621
5	6	0.0290058511	0.0354074663741
6	9	-0.0304069342	-0.0177439150251
7	10	-0.0665061285	-0.0595950656119
8	11	-0.0667191889	-0.0446611130085
9	12	-0.0101262400	-0.0110180170713
10	13	0.0038518768	0.0054364685667
11	Angle of Attack	2.70	2.24

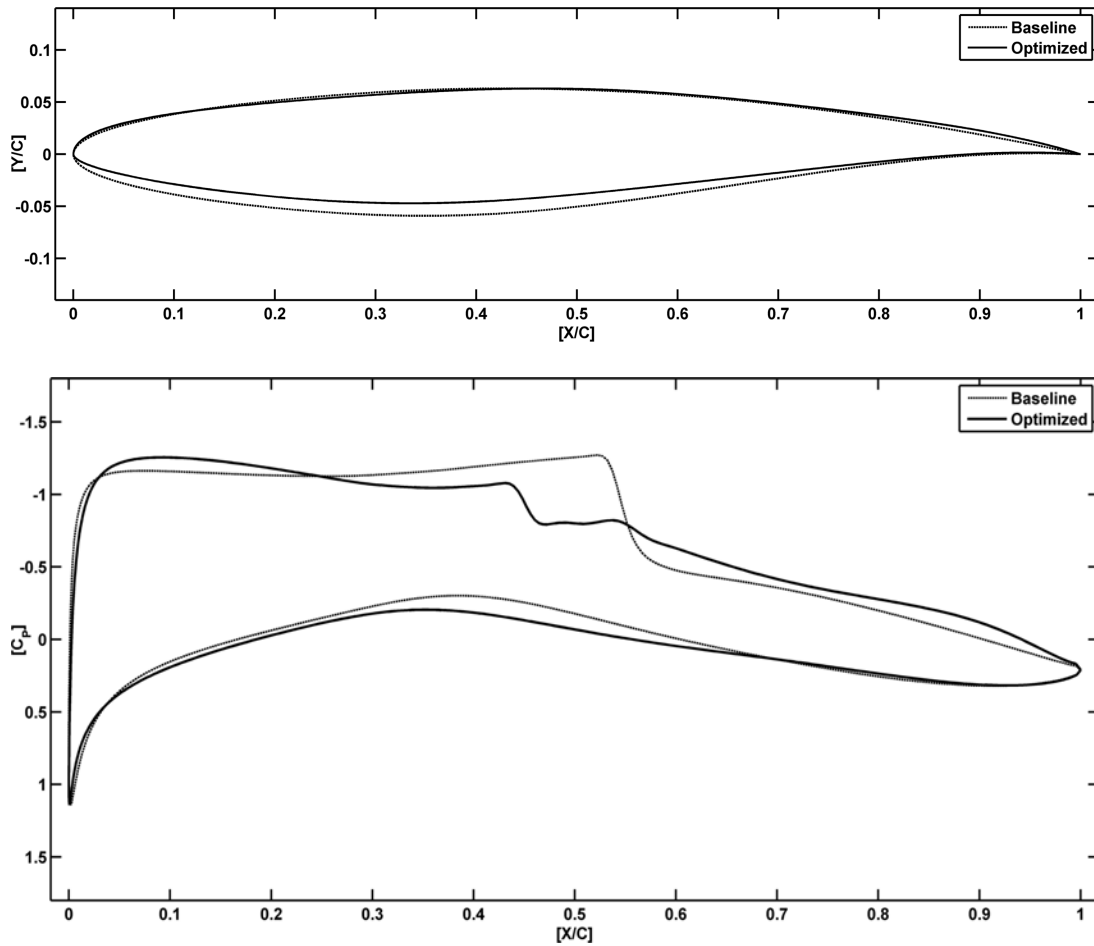


Figure 25 RAE 2822 Baseline and Optimized airfoil shapes C_p distributions

From the surface C_p distributions it can be seen that the adverse pressure gradient has been significantly reduced, resulting in an almost-shock free airfoil. This has also resulted in reduced overall drag.

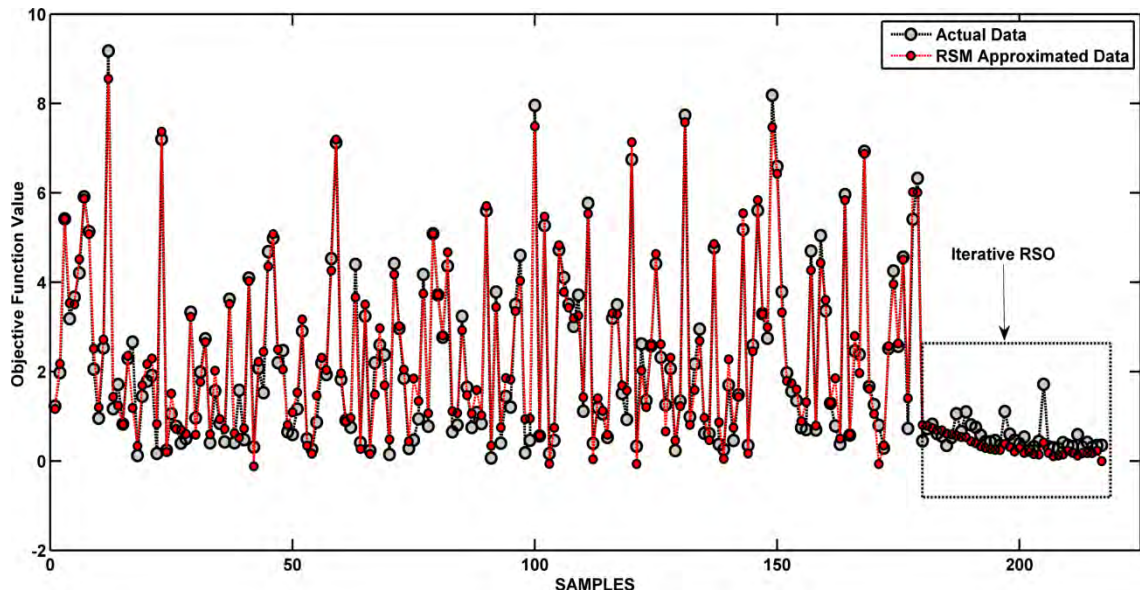


Figure 26 Plot comparing the actual and predicted function values

Fitness Parameters

- **Adjusted root mean square error σ_a**

$$\sigma_a = 0.467095$$

This is a small value compared to the data. Thus the model is good enough.

- **Coefficient of multiple determination R_{adj}^2**

$$R_{adj}^2 = 0.94259$$

This value of R_{adj}^2 is close to 1. Thus it indicates a good model fit.

4.3 Adaptive Airfoil Design based on the Boeing-737C airfoil

The iterative RSO methodology is employed for the optimization of a Boeing-737-300 Classic airfoil, taking it as a representative case in an attempt to demonstrate the advantages of using an adaptive airfoil over a conventional fixed airfoil.

The airfoil is of a near-symmetric shape and is one of 737's inboard mid-span airfoil, located between the engine and the aircraft fuselage.

The Maximum Take-Off weight of the aircraft is 62,820[kg] and the total wing planform area is 105.4 [m²]. It has a cruising speed of Mach 0.74 at 35,000[ft] (Service ceiling of 37,000 [ft]).

4.3.1 Operational Envelope

The flight envelope or performance envelope of an aircraft refers to the capabilities of a design in terms of airspeed and load factor or altitude. A doghouse plot generally shows the relation between speed at level flight and altitude although other variables are also possible. The plot typically looks something like an upside-down U and is commonly referred to as a doghouse plot due to its resemblance to a doghouse. A typical flight envelope for a subsonic (or transonic) aircraft is shown below:

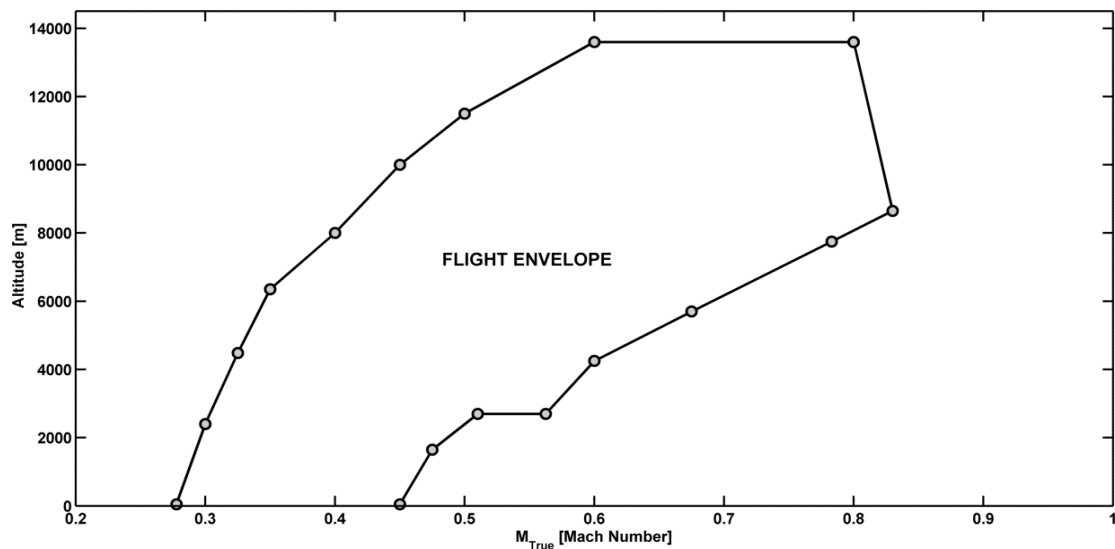


Figure 27 Typical aircraft flight envelope

The outer edges of the diagram, the envelope, show the possible conditions that the aircraft can reach in straight and level flight. The low (stall) speed boundary indicates that the aircraft wing can only produce enough lift for a given speed at various altitudes. The stalling speed (in terms of true airspeed) increases with altitude (as the density reduces). The service ceiling is the maximum altitude that the aircraft attains while climbing at a very small climb rate (typically 100 [ft/min]. according to FAR Part 25 regulations).

In the maximum operating Mach number region, the aircraft reaches a region of flight where the drag produced increases sharply (i.e., drag divergence Mach number boundary) and thus the aircraft engines are incapable of producing enough thrust to accelerate the aircraft to faster speeds. The design of all aircraft structures carries an assumption about the maximum loads that can be tolerated in flight. This is limited by the dynamic pressure limit boundary. The bird-strike limit boundary is

defined by taking into account the bird-hits on the wind-shields below 10,000[ft]. Traditionally this boundary has been set at 250 knots.

4.3.2 C_L requirements across the envelope

The flight envelope is discretized into discrete points, in that various altitude-Mach number points are obtained within the flight envelope. Given a specific load requirement [per unit area of the wing], the required values of the lift coefficients are obtained at these points. This is as shown below—

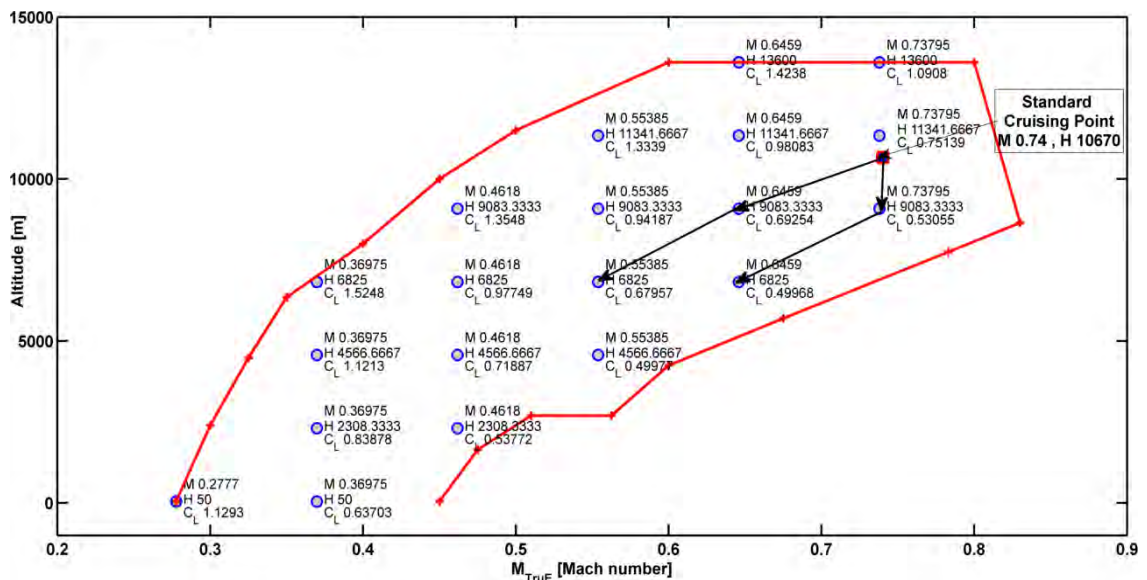


Figure 28 Discretized points in the flight envelope.

Here the region bounded by the flight envelope is discretized uniformly and the various operating points are obtained. However, only those operating points are chosen for the purpose of optimization, which have a greater probability of being visited by a typical subsonic transport airplane. The standard cruising point is Mach 0.74 and altitude H= 10670[m]. So it is assumed that the baseline airfoil has been optimized for this operating point. For the adaptive airfoils problem, three off-design points are chosen—

- A) Mach 0.74 and altitude H = 9083.3 [m]
- B) Mach 0.65 and altitude H = 9083.3 [m]
- C) Mach 0.65 and altitude H = 6825.0 [m]

The corresponding values for the C_L at these points are—A) 0.5305, B) 0.6925, and C) 0.4996. These values are obtained using the lift force obtained from the airfoil at the standard cruising condition, which is ≈ 6150 [N].

4.3.3 Airfoil Geometry

The airfoil geometry for the Boeing airfoil is obtained from the airfoil database which has been published on the University of Illinois-Urbana Campaign (UIUC) website. It is a near-symmetric airfoil, with very little rear-loading and strong shockwaves exist on the upper-surface of the airfoil near the leading edge.

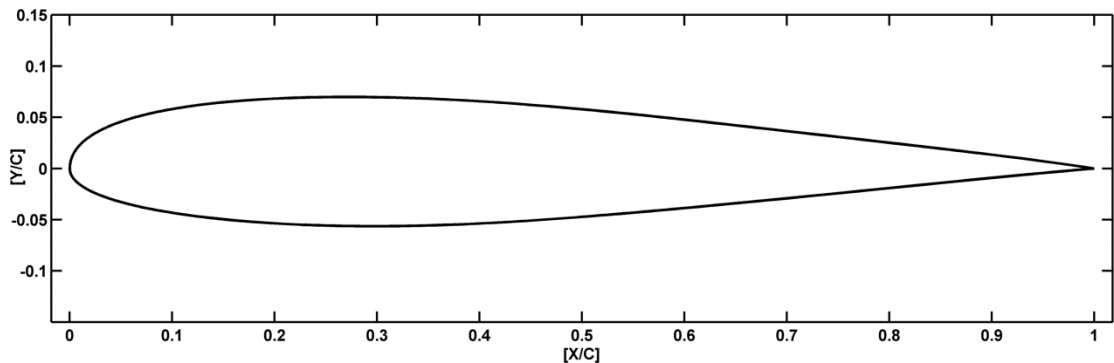


Figure 29 Boeing 737 airfoil (baseline)

The parameterization for this airfoil is performed using B-spline curve fitting consisting of 10 control points (5 for each surface). This is as shown below

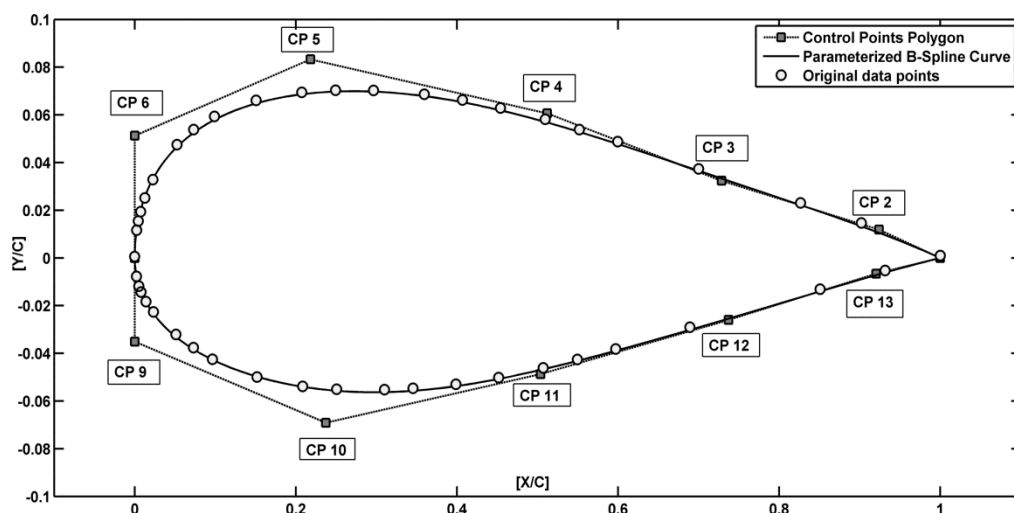


Figure 30 B-Spline parameterization of the Boeing 737 airfoil (expanded)

The range for each DV, as before was obtained by defining an upper and a lower limit. However, in this case the upper and lower limits were defined by values 25% above and below the baseline values. This was done so that shapes differing much from the baseline shape are not attained and investigated during the optimization process. Thus the required deformations from the baseline shape to the optimized shape would be less.

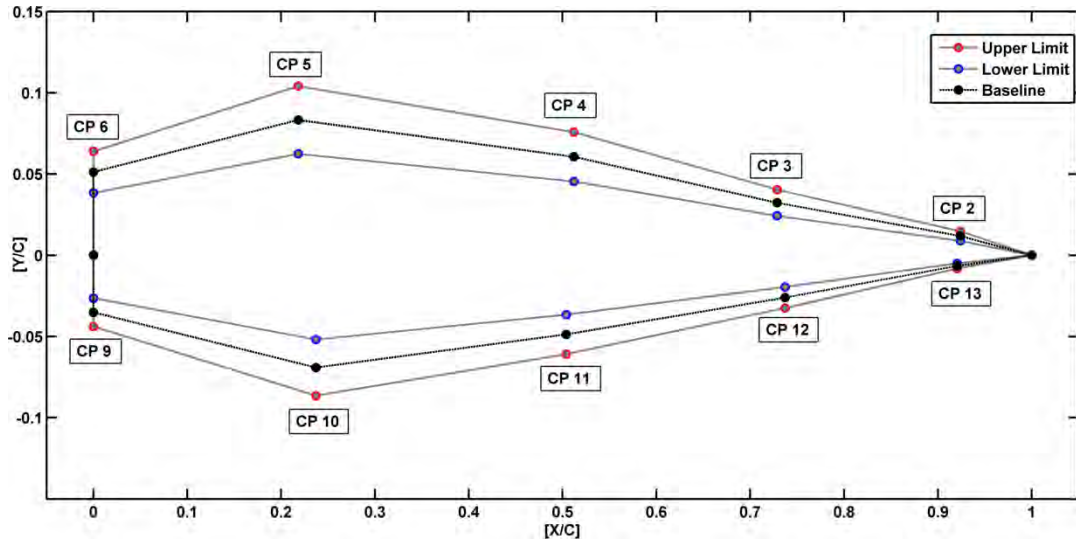


Figure 31 Range of the Boeing-737 airfoil design variables

The baseline, and the upper and lower limit values of the design variables are as follows—

Table 6 Baseline and Upper and Lower limit Values of the Design Variables

DV	CP	Baseline Values	Upper Limit	Lower Limit
1	2	0.01189154138	0.014864426725	0.0089186560354
2	3	0.03237346633	0.040466832919	0.0242800997514
3	4	0.06064856760	0.075810709508	0.0454864257052
4	5	0.08333191253	0.104164890663	0.0624989343979
5	6	0.05126309035	0.064078862946	0.0384473177681
6	9	-0.03508193513	-0.043852418920	-0.0263114513520
7	10	-0.06910529131	-0.086381614144	-0.0518289684866
8	11	-0.04869035929	-0.060862949114	-0.0365177694684
9	12	-0.02601627424	-0.032520342809	-0.0195122056855
10	13	-0.00657293832	-0.008216172907	-0.0049297037445
11	Angle of Attack	2.5 to 3.5	1.5	4.0

4.3.4 CASE-A: Mach 0.74 at 9083.3 [m] (29,800[ft])

This corresponds to the off-design operating condition where the aircraft is cruising in steady flight at Mach 0.74 (True airspeed) and 9083.33[m] (29800[ft]) above ground level). Following are the freestream conditions for this case—

Table 7 Freestream conditions for the off-design operating point A

True Airspeed	Mach 0.74 [-]
	224.4616 [m/s] (@9083.04[m])
Altitude	9083.3 [m] (29,800 [ft])
Atmospheric Pressure	30363.99 [N/m ²]
Air Density	0.461696 [kg/m ³]
Ambient Temperature	229.12024 [K] (-43.879 [°C])
Dynamic Viscosity	1.7894e-05 [kg/m-s]
Reynolds Number	5.791x10 ⁶ [-]
Angle of Attack	2.62712°

Following are the values of the optimum design variables for the operating point-A—

Table 8 Baseline and Optimized values of the Design Variables

DV	CP	Baseline Values	Optimum Values
1	2	0.01189154138	0.0094258191179
2	3	0.03237346633	0.0265314506846
3	4	0.06064856760	0.0716335747285
4	5	0.08333191253	0.066898113326
5	6	0.05126309035	0.0467090526596
6	9	-0.03508193513	-0.0283241637483
7	10	-0.06910529131	-0.0710298081602
8	11	-0.04869035929	-0.0395037365250
9	12	-0.02601627424	-0.023849412116
10	13	-0.00657293832	-0.0071745014079
11	Angle of Attack	2.627°	2.209°

The following values compare the performance parameters (Lift, moment and drag) of the baseline and optimized airfoil. Here (C-W) refers to Clockwise moment and (C-CW) refers to counter-clockwise moments—

Table 9 Baseline and Optimized performance at off-design operating point

PERFORMANCE PARAMETERS	BASELINE	OPTIMIZED	REMARKS
LIFT FORCE			
Pressure Lift [N]	6141.1205	6153.514	
Viscous Lift [N]	0.709057	0.09236	
Total Lift [N]	6141.8296 [REQUIRED]	6153.6064	DIFFERENCE OF 0.19% (HIGHER)
Pressure CL [-]	0.5280	0.529069	
Viscous CL [-]	6.0965e-05	7.94118e-06	
Total CL [-]	0.52806 [REQUIRED]	0.529077	DIFFERENCE OF 0.19% (HIGHER)
MOMENTS			
Pressure Moment [N-m]	175.5258	150.5031	
Viscous Moment [N-m]	0.15634	-0.41194	
Total Moment [N-m]	175.6822 [C-CW] [REQUIRED]	150.09122 [C-CW]	DIFFERENCE OF 14.5% (LESS C-CW)
Pressure CM [-]	-0.01509	0.01294	
Viscous CM [-]	1.34421e-05	-3.5418e-05	
Total CM [-]	0.015104 [C-CW] [REQUIRED]	0.012904 [C-CW]	DIFFERENCE OF 14.5% (LESS C-CW)
DRAG FORCE			
Pressure Drag [N]	375.8002	169.5884	
Viscous Drag [N]	55.4470	62.1306	
Total Drag [N]	431.2472	231.71912	REDUCTION OF 46.26%
Pressure CD [-]	0.03231	0.01458	
Viscous CD [-]	0.004767	0.0053419	
Total CD [-]	0.0370779	0.0199228	REDUCTION OF 46.26%
$\Delta CD = -0.017155 \approx 17$ [DRAG COUNTS] LESS			
ANGLE OF ATTACK	2.6271o	2.209o	

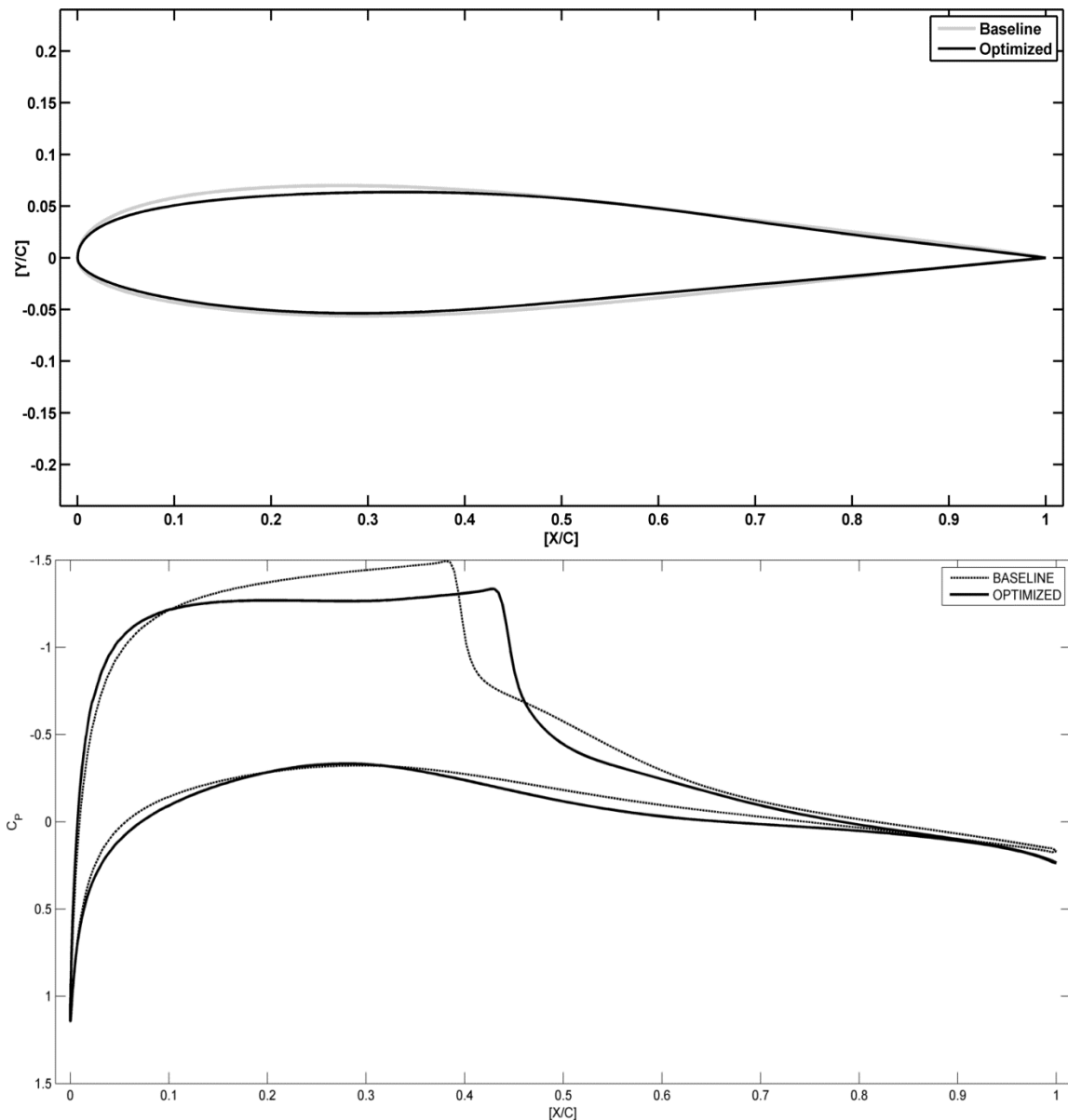


Figure 32 CASE-A Baseline and Optimized airfoil shapes and their corresponding surface C_p distributions

From the C_p distribution it can be seen that the shock intensity has been reduced and that supposedly the transition has been delayed, thus reducing shock induced drag.

Model Evaluation

The RSM is constructed using 200 samples as in the DoE. Following this, 26 iterative improvements are performed on the RSM. Following are the RSM testing values and a comparison between the actual values of the objective function and the predicted values by the RSM—

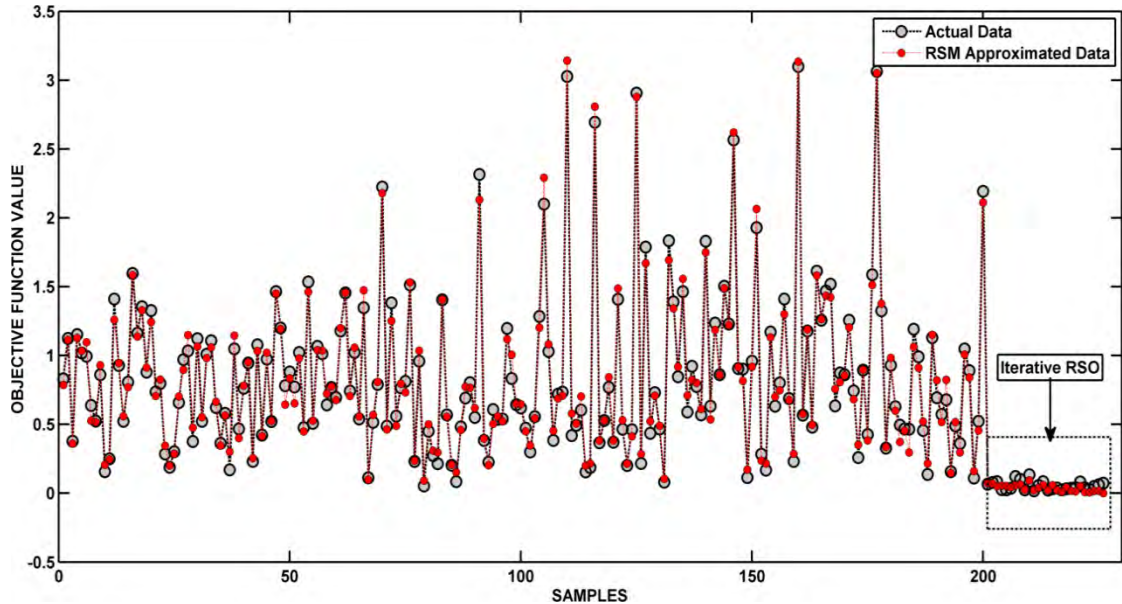


Figure 33 Plot comparing the actual and the predicted objective function values (CASE-A)

Fitness Parameters

- **Root Mean Square Error (RMSE)**

$$\text{RMSE} = 6.3575\%$$

As the value of the RMSE is <10%, this indicates a reasonably good model

- **Adjusted root mean square error σ_a**

$$\sigma_a = 0.6145$$

This is a small value compared to the data. Thus the model is good enough.

- **Coefficient of multiple determination R_{adj}^2**

$$R_{adj}^2 = 0.9831$$

This value of R_{adj}^2 is close to 1. Thus it indicates a good model fit.

4.3.5 CASE-B: Mach 0.65 at 9083.3 [m] (29,800[ft])

This corresponds to the off-design operating condition where the aircraft is cruising in steady flight at Mach 0.65(True airspeed) and 9083.33[m] (29800[ft] above ground level). Following are the freestream conditions for this case—

Table 10 Freestream conditions for the off-design operating point B

True Airspeed	Mach 0.65 [-]
	197.1622 [m/s] (@9083.04[m])
Altitude	9083.3 [m] (29,800 [ft])
Atmospheric Pressure	30363.99 [N/m ²]
Air Density	0.461696 [kg/m ³]
Ambient Temperature	229.12024 [K] (-43.879 [°C])
Dynamic Viscosity	1.7894e-05 [kg/m-s]
Reynolds Number	5.087x10 ⁶ [-]
Angle of Attack	4.0271 °

Following are the values of the optimum design variables for the operating point-A—

Table 11 Baseline and Optimized values of the Design Variables

DV	CP	Baseline Values	Optimum Values
1	2	0.01189154138	0.010564514738021
2	3	0.03237346633	0.025643292357113
3	4	0.06064856760	0.072880447979943
4	5	0.08333191253	0.082824625968899
5	6	0.05126309035	0.042007873882886
6	9	-0.03508193513	-0.035352800149791
7	10	-0.06910529131	-0.058162753224287
8	11	-0.04869035929	-0.053120614103188
9	12	-0.02601627424	-0.030421540795948
10	13	-0.00657293832	-0.006672806068045
11	Angle of Attack	4.0271°	3.8795°

The following values compare the performance parameters (Lift, moment and drag) of the baseline and optimized airfoil. Here (C-W) refers to Clockwise moment and (C-CW) refers to counter-clockwise moments—

Table 12 Baseline and Optimized performance at off-design operating point B

PERFORMANCE PARAMETERS	BASELINE	OPTIMIZED	REMARKS
LIFT FORCE			
Pressure Lift [N]	6141.1205	6189.2509	
Viscous Lift [N]	0.709057	-0.06223	
Total Lift [N]	6141.8296 [REQUIRED]	6189.1887	DIFFERENCE OF 0.77% (HIGHER)
Pressure C_L [-]	0.68434	0.689706	
Viscous C_L [-]	7.9015e-05	-6.935388e-06	
Total C_L [-]	0.68442 [REQUIRED]	0.6897	DIFFERENCE OF 0.77% (HIGHER)
MOMENTS			
Pressure Moment [N-m]	-39.8508	-35.6979	
Viscous Moment [N-m]	-0.6048	-0.9138	
Total Moment [N-m]	-40.4556 [C-W] [REQUIRED]	-36.6117 [C-W]	DIFFERENCE OF 9.5% (LESS C-W)
Pressure C_M [-]	-0.0044	-0.003978	
Viscous C_M [-]	-6.73958e-05	-0.0001018	
Total C_M [-]	-0.004508 [C-W] [REQUIRED]	-0.004079 [C-W]	DIFFERENCE OF 9.5% (LESS C-W)
DRAG FORCE			
Pressure Drag [N]	157.172	70.82073	
Viscous Drag [N]	46.0286	50.04897	
Total Drag [N]	203.2007	120.8697	REDUCTION OF 40.5%
Pressure C_D [-]	0.017515	0.0078919	
Viscous C_D [-]	0.005129	0.0055772	
Total C_D [-]	0.022643	0.0134692	REDUCTION OF 40.5%
$\Delta C_D = -0.009174 \approx 9$ [DRAG COUNTS] LESS			
ANGLE OF ATTACK	4.0271°	3.8795°	

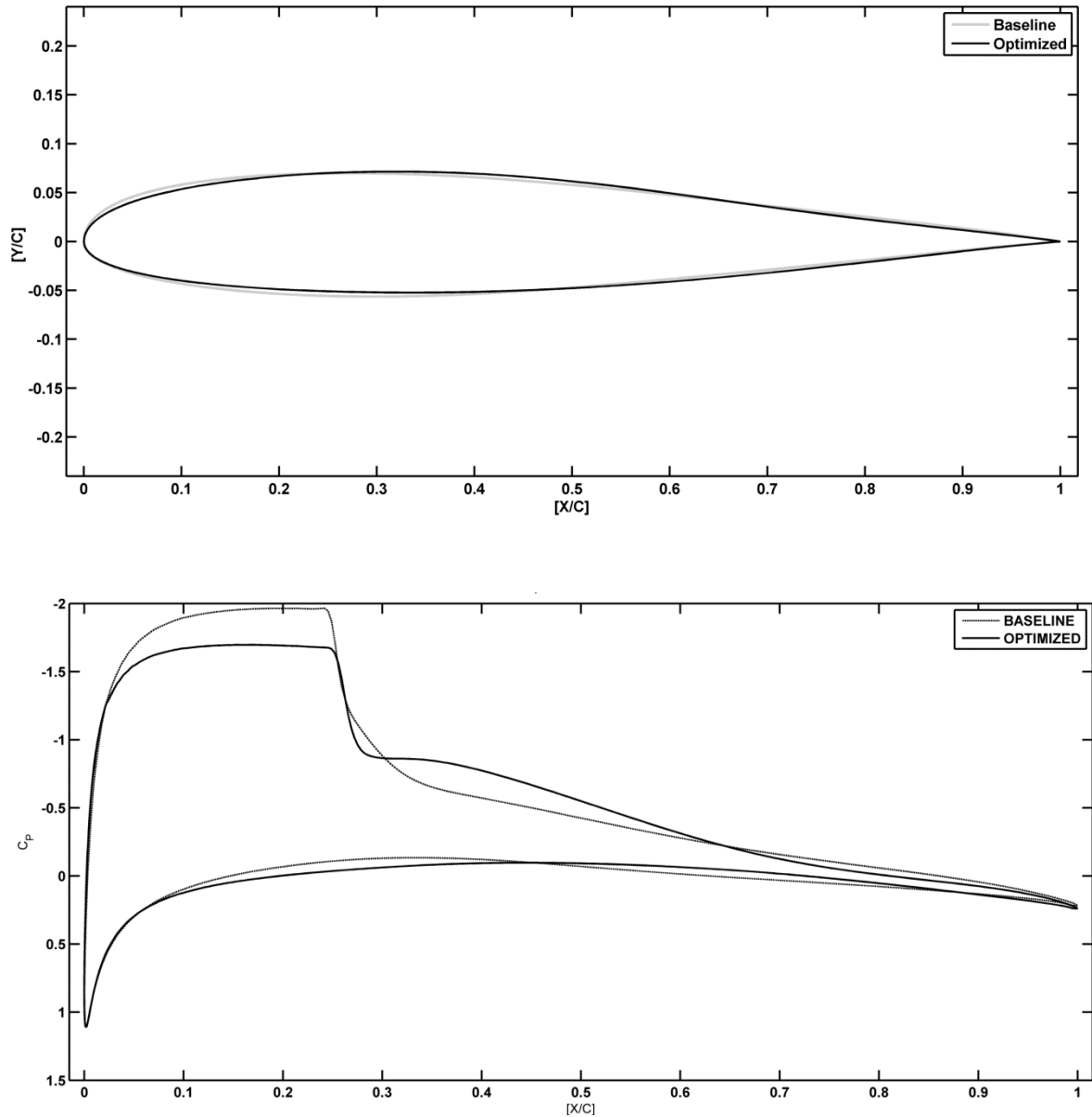


Figure 34 CASE-B Baseline and Optimized airfoil shapes and their corresponding surface C_p distributions

From the C_p distribution it can be seen that the shock intensity has been reduced the turbulence has not been delayed much, the reduction shock induced drag seems to have contributed to the overall drag reduction significantly.

Model Evaluation

The RSM is constructed using 200 samples as in the DoE. Following this, 26 iterative improvements are performed on the RSM. Following are the RSM testing values and a comparison between the actual values of the objective function and the predicted values by the RSM—

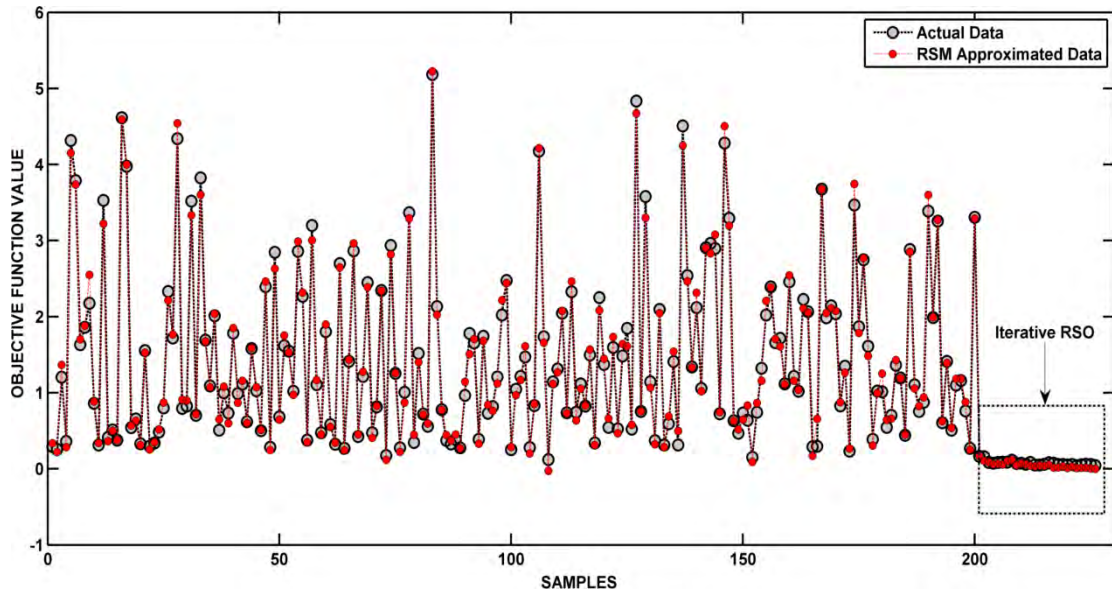


Figure 35 Plot comparing the actual and the predicted objective function values (CASE-B)

Fitness Parameters

- **Root Mean Square Error (RMSE)**

$$\text{RMSE} = 9.995\%$$

As the value of the RMSE is **<10%**, this indicates a reasonably good model

- **Adjusted root mean square error σ_a**

$$\sigma_a = 0.1235$$

This is a small value compared to the data. Thus the model is good enough.

- **Coefficient of multiple determination R_{adj}^2**

$$R_{adj}^2 = 0.9885$$

This value of R_{adj}^2 is close to 1. Thus it indicates a good model fit.

4.3.6 CASE-C: Mach 0.65 at 6827.0 [m] (22,400[ft])

This corresponds to the off-design operating condition where the aircraft is cruising in steady flight at Mach 0.65(True airspeed) and 6827[m] (22400[ft] above ground level). Following are the freestream conditions for this case—

Table 13 Freestream conditions for the off-design operating point C

True Airspeed	Mach 0.65 [-]
	203.45 [m/s] (@6827.0[m])
Altitude	6827.0 [m] (22,400 [ft])
Atmospheric Pressure	42067.06 [N/m ²]
Air Density	0.6012[kg/m ³]
Ambient Temperature	243.7811[K] (-29.368[°C])
Dynamic Viscosity	1.7894e-05 [kg/m-s]
Reynolds Number	6.835x10 ⁶ [-]
Angle of Attack	2.55°

Following are the values of the optimum design variables for the operating point-A—

Table 14 Baseline and Optimized values of the Design Variables C

DV	CP	Baseline Values	Optimum Values
1	2	0.01189154138	0.0128883193059
2	3	0.03237346633	0.0275768561427
3	4	0.06064856760	0.0714721803511
4	5	0.08333191253	0.0689957016386
5	6	0.05126309035	0.0385565064030
6	9	-0.03508193513	-0.0283429759241
7	10	-0.06910529131	-0.0667506140691
8	11	-0.04869035929	-0.0503457197813
9	12	-0.02601627424	-0.0297665255502
10	13	-0.00657293832	-0.00786047883110
11	Angle of Attack	2.55°	2.7346884

The following values compare the performance parameters (Lift, moment and drag) of the baseline and optimized airfoil. Here (C-W) refers to Clockwise moment and (C-CW) refers to counter-clockwise moments—

Table 15 Baseline and Optimized performance at off-design operating point C

PERFORMANCE PARAMETERS	BASELINE	OPTIMIZED	REMARKS
LIFT FORCE			
Pressure Lift [N]	6141.1205	6066.573	
Viscous Lift [N]	0.709057	-0.56088	
Total Lift [N]	6141.8296 [REQUIRED]	6066.01273	DIFFERENCE OF 1.23% (LOWER)
Pressure C_L [-]	0.4936	0.4879	
Viscous C_L [-]	5.698e-05	-4.511e-05	
Total C_L [-]	0.4936 [REQUIRED]	0.487917	DIFFERENCE OF 1.23% (LOWER)
MOMENTS			
Pressure Moment [N-m]	57.92965	97.35203	
Viscous Moment [N-m]	-1.1563	-0.99956	
Total Moment [N-m]	56.7733 [C-CW] [REQUIRED]	96.3524	DIFFERENCE OF 69.71% (MORE C-CW)
Pressure C_M [-]	0.004655	0.0078604	
Viscous C_M [-]	-9.2931e-05	-8.039e-05	
Total C_M [-]	0.00456 [C-CW] [REQUIRED]	0.00775	DIFFERENCE OF 69.71% (MORE C-W)
DRAG FORCE			
Pressure Drag [N]	63.479	44.9669	
Viscous Drag [N]	70.80876	72.6274	
Total Drag [N]	134.28784	117.59436	REDUCTION OF 12.43%
Pressure C_D [-]	0.0051059	0.003616	
Viscous C_D [-]	0.0056954	0.005841	
Total C_D [-]	0.0108013	0.009458	REDUCTION OF 12.43%
$\Delta C_D = -1.345 \approx 1.5$ [DRAG COUNTS] LESS			
ANGLE OF ATTACK	2.55°	2.735°	

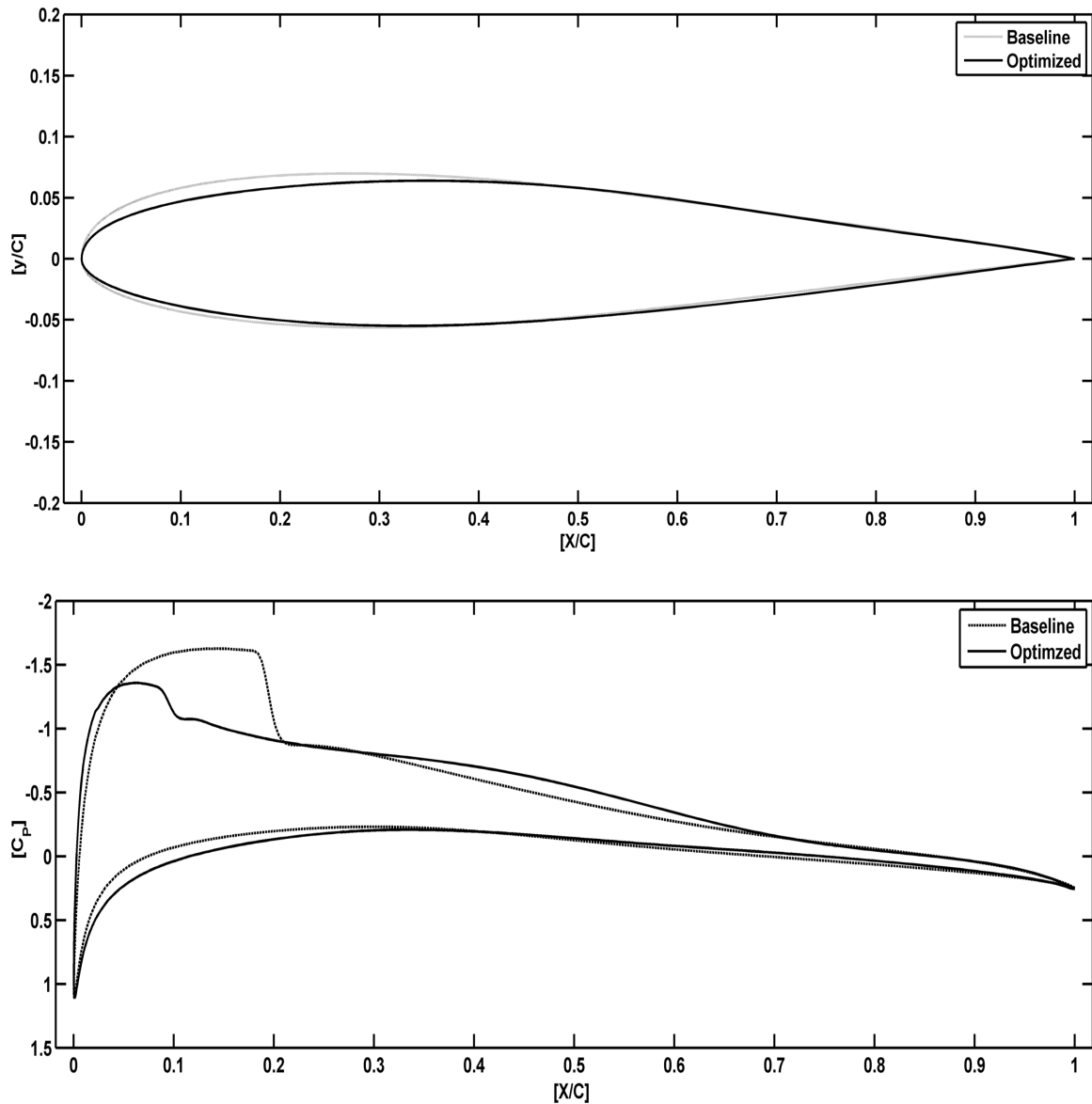


Figure 36 Plot comparing the actual and the predicted objective function values (CASE-C)

From the C_p distribution it can be seen that the shock intensity has drastically been reduced when compared to the baseline airfoil. Thus the shock induced drag has been reduced. However the lift generated is slightly less when compared to the required lift.

Model Evaluation

The RSM is constructed using 200 samples as in the DoE. Following this, 26 iterative improvements are performed on the RSM. Following are the RSM testing values and a comparison between the actual values of the objective function and the predicted values by the RSM—

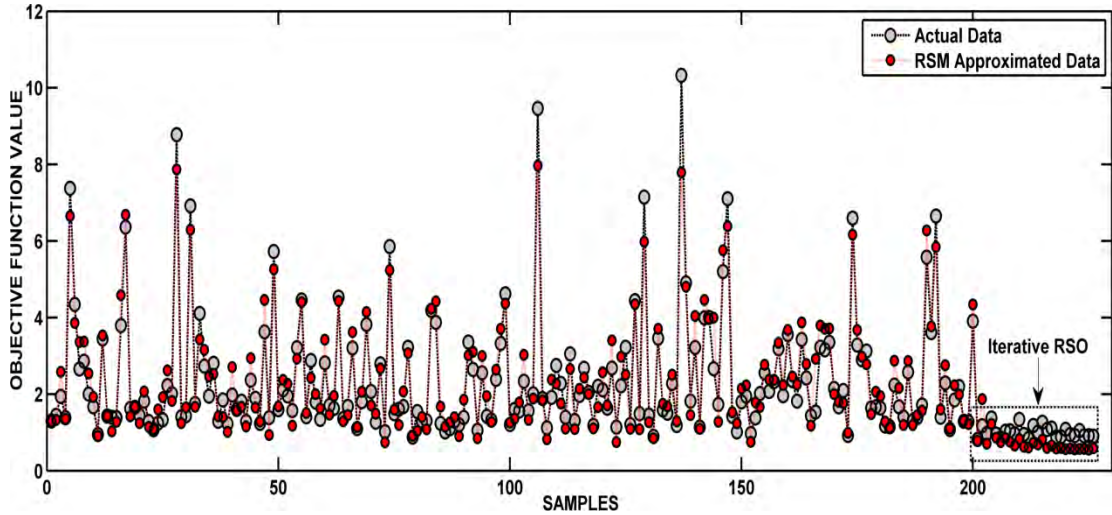


Figure 37 Plot comparing the actual and the predicted objective function values (CASE-C)

Fitness Parameters

- **Adjusted root mean square error σ_a**

$$\sigma_a = \mathbf{0.53410}$$

This is a small value compared to the data. Thus the model is good enough.

- **Coefficient of multiple determination R_{adj}^2**

$$R_{adj}^2 = \mathbf{0.8873760}$$

This value of R_{adj}^2 is close to 1. Thus it indicates a decent model fit.

4.4 Performance Comparison of Baseline Airfoil Relative to Optimized Airfoil

Additional flow simulations are performed to assess the performance improvement (drag reduction) of the optimized airfoil relative to the baseline airfoil. This is done on the basis of three criteria as follows—

- Fully optimized airfoil (Both optimized shape and angle of attack).
- Baseline airfoil shape at optimized angle of attack.
- Optimized airfoil shape at baseline angle of attack.

This analysis is done for the all the case studies discussed above— RAE 2822 Design study, Boeing 737C (Cases A, B and C). The change in drag is computed relative to the baseline airfoil drag

Table 16 Performance Comparison Analysis

CASES	Particulars	Baseline Airfoil	Performance Comparison Criteria		
			Fully Optimized Airfoil	Baseline Airfoil Shape @ Optimized AoA	Optimized Airfoil Shape @ Baseline AoA
RAE-2822 Design Study	DRAG [N]	270.9082	206.989	211.9949	281.4196
	AoA	2.7	2.24	2.24	2.7
	Drag Change	-	-23.59%	-21.74	+3.88%
Boeing 737C CASE-A M 0.74, H 29800ft	DRAG [N]	431.2472	213.7191	360.9521	309.6035
	AoA	2.627	2.209	2.209	2.627
	Drag Change	-	-46.26%	-16.29%	-28.2%
Boeing 737C CASE-B M 0.65, H 29800 ft	DRAG [N]	203.2007	120.8697	187.8078	129.9117
	AoA	4.027	3.879	3.879	4.027
	Drag Change	-	-40.5%	-7.5%	-36.06%
Boeing 737C CASE-C M 0.65, H 22400 ft	DRAG [N]	134.2878	117.5943	144.0607	115.2182
	AoA	2.55	2.735	2.735	2.55
	Drag Change	-	-12.43%	+7.27%	-14.2%

AoA → Angle of Attack

4.5 Adaptive Airfoils

The adaptive airfoil is designed by optimizing the shape for the 3 cases (A, B and C) as discussed above. It can be seen from the above results that the individual single point optimized airfoils have less drag than the baseline drag. A maximum of 46% reduction is achieved using the adaptive airfoil for a particular operating point. This is a significant drag reduction considering the amount of variation in the shape required

Maximum movement of the upper surface is roughly 1.1% chord while the maximum lower surface movement is roughly 0.3% chord. The maximum variation in the lift force generated is 1.23%.

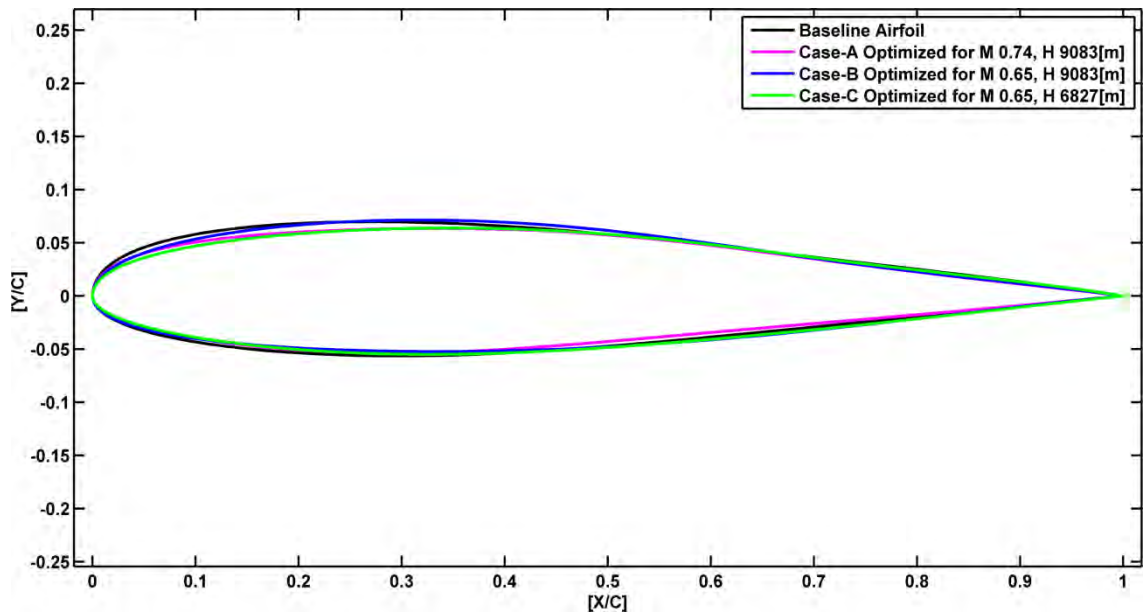


Figure 38 Baseline and Optimized Airfoil shapes for Cases A, B, and C

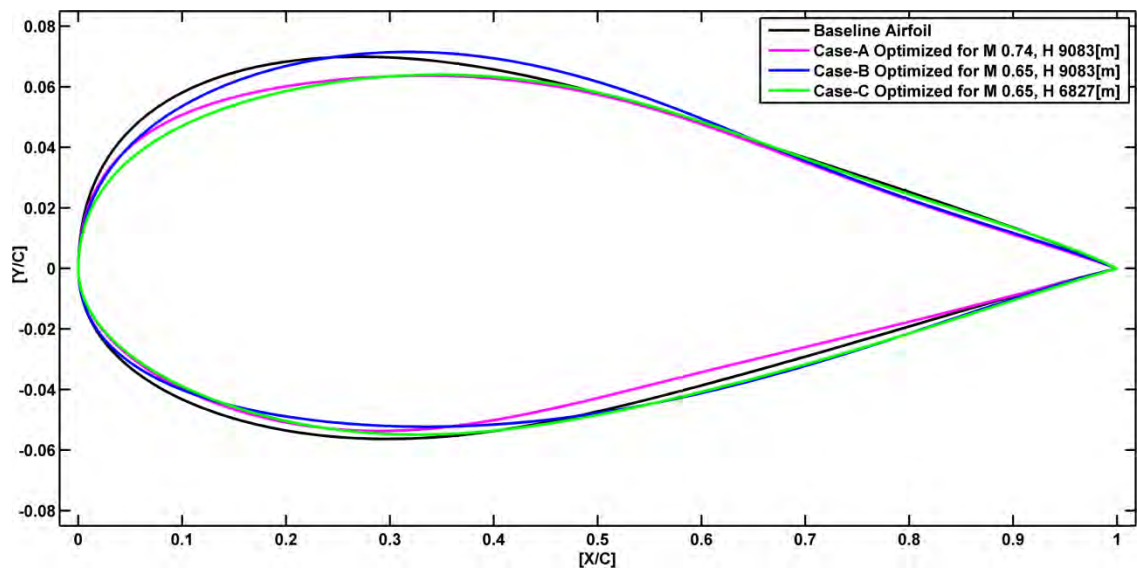


Figure 39 Baseline and Optimized Airfoil shapes for Cases A, B, and C (Expanded View)

Note that the adaptive concept can be utilized in two different ways. When operating at a given Mach number, the shape can be altered to match the airfoil optimized at the Mach number closest to the given Mach number. Alternatively, one can determine a continuous variation of shape throughout the Mach number range.

5. CONCLUSION AND FUTURE WORK

5.1 Summary and Conclusion

In this research, an iterative response surface optimization methodology is employed for carrying out aerodynamic shape optimization on various case studies. These include a design validation study on the RAE 2822 airfoil and the shape optimization of a Boeing-737 classic airfoil at various off-design operating conditions.

The parameterization is carried out using B-spline control points, the y-ordinates of which, and the angle of attack are taken as the design variables for the optimization process. Using the results obtained from CFD simulations, a full quadratic polynomial response surface model (RSM) is constructed, which is then optimized using the SQP technique to obtain the optimum values of the design variables. For constructing the RSM, the Latin Hypercube Sampling (LHS) design is used to obtain the Design of Experiments (DoE) plan. Polynomial regression is used to obtain the values of the regression coefficients. The RSM is improved in an iterative manner, in that a CFD simulation is performed for the optimum values of the design variables obtained at the end of each optimization step and the resulting value of the objective function (actual value) along with the design variable values are added to the initial DoE plan. Polynomial regression is performed again and new values of the regression coefficients are obtained which are then used to re-construct the RSM. This process is repeated till the difference between the actual and approximated objective function value (value obtain from the RSM) does not lie within a tolerance limit.

The main advantage of using this approach for shape optimization problems is that values obtained from commercially available flow solvers can directly be used in the optimization process, without making any changes to the solver's code. Also the noise and non-smoothness issues associated with CFD results are smoothed out by using the RSM which is quadratic polynomial in terms of the design variables. Thus the optimization process can be performed effectively and smoothly without any sudden divergence issues associated with the CFD results.

The mesh required for the simulation is generated using GridPro which is an automatic, topology based multi-block structured grid generator with an advanced

grid smoothening algorithm. ANSYS-FLUENT is used as the flow solver to perform the CFD simulation. Here the governing flow equations, the continuity, Navier-Stokes and the energy conservation equations are solved using a pressure-based solver. In addition, the Spalart-Allmaras turbulence model, which is a one equation fully turbulent model, is used to close the momentum equations. The results obtained from the CFD simulations are the lift, drag and moment coefficients of the airfoil given the free-stream conditions. These values are then used to evaluate the objective function value.

The objective function used here is the lift-moment constrained drag minimization problem. It is a penalty augmented type problem in which the constraints (lift and moment coefficients) are added to the objective function as a quadratic penalty term. Doing so converts the constrained problem into an unconstrained problem and thus there is no requirement for the evaluation of the constraints and their gradients separately.

MATLAB Optimization and Global Optimization Toolbox functions are used to perform the optimization process. The main tasks performed using MATLAB are—the parameterization of the airfoil shape, the DoE plan, RSM construction and the iterative RSO processes. `fmincon` and `MultiStart` solvers are employed using the SQP algorithm to optimize the RSM and obtain values of the optimum design variables.

As mentioned earlier, the iterative RSO methodology has been used here for the aerodynamic shape optimization of various airfoils. These include—the RAE 2822 airfoil as a design validation study, and the Boeing-737 airfoil at 4 different off-design operating conditions.

The Boeing 737 classic airfoil is optimized at four off-design operating conditions. This has been used as a representative case-study to demonstrate the advantages of using an adaptive airfoil over a fixed geometry airfoil. The results obtained are encouraging in lieu of the fact that significant drag reduction is achieved by morphing the airfoil shape from one operating condition to the other. The baseline drag at off-design conditions is found to be significantly higher than that of the optimized adaptive airfoil shape at the same condition. The drag-reduction is attributed primarily to the reduction in the intensity of the shock wave on the upper surface of the airfoil. Also the location of the adverse pressure gradient is pushed back

downstream (towards the trailing edge), thus delaying the transition from laminar and turbulent. This is achieved by morphing the baseline geometry to a better suited one which is obtained from the aerodynamic shape optimization process. The optimized shape offers lesser drag while maintaining the desired lift force and the baseline pitching moment. The idea behind retaining the baseline pitching moment is that, for an optimized geometry, if there is a change in the pitching moment of the airfoil, the aircraft elevators would have to be deployed to account for this change. This indirectly would offset the benefit obtained from morphing the airfoil.

About 40% drag reduction is achieved by implementing the aerodynamic shape optimization methodology that has been used in this research. The maximum change in the geometry (from baseline to optimized) is about 0.2% of the airfoil chord-length which is relatively very small compared to the overall dimensions of the airfoil. This implies that very little deformations are required to obtain aerodynamically efficient airfoils, which offer significantly lesser drag over a wider portion of the flight envelope compared to fixed geometry airfoils which have been optimized for only few operating points in the envelope.

Thus the adaptive airfoil concept is a viable concept which can be pursued with confidence. Given the fact that little deformations are required to obtain much efficient airfoil shapes, the adaptive airfoil concept can be considered feasible from the structural point of view as well.

However there are a lot of issues that remain to be addressed before this technology can be implemented. Even though the required deformations are small, there are structural challenges that hamper the full-scale development of this technology. The ideal material that can be used as the skin for adaptive airfoils should have high strength, can withstand huge strains without permanent deformation with the ability to regain the original shape precisely, low stiffness, excellent fatigue properties and low weight. Such a material is central to the development of this concept. But as of now this area is still under-research. In addition to materials, robust mechanisms have to be developed that can deform the skin to the desired (optimized) shapes. Such mechanisms should be feasible, not involve a very complex architecture and require very little actuation energy. A control system is then required that can sense the change in the aircraft operating condition, and provides the necessary signals that can actuate the mechanisms to obtain the desired shape.

5.2 Future Work

There remains a lot of scope for improvement in the optimization methodology that has been employed in this research work. In particular the flow solver and the turbulence model that has been used here can be replaced with higher-fidelity flow solvers that can predict the flow transition from laminar to turbulent more accurately. Laminar-turbulent flow transition prediction plays a vital role in the accuracy of the results, particularly in drag prediction. However the most commonly used turbulence model do not predict the transition well, and thus they can be replaced by other models such as the *k-kl-w Transition* or the *Transition SST* models.

Also Large-Eddy Simulation (LES), in which the large scale eddies (turbulence) are filtered and resolved, can be employed instead of the RANS equations in which the eddies are time averaged. Thus more accurate results can be obtained.

For the optimization of the response surface model, Genetic Algorithms or Particle Swarm Optimization techniques can be investigated to obtain the optimum values of the design variables, instead of the gradient based optimization method (SQP) that has been used here. Using Genetic Algorithms and other similar techniques can improve the chances of locating a global minimum instead of the local minimum that is obtained using gradient based methods.

In this work, the methodology has been applied for the optimization of an airfoil which is basically a 2-dimensional shape. Even though having the right airfoil shape is essential, the 3-dimensional shape of the wing is equally crucial. A lot of other factors need to be addressed while designing the 3-dimensional shape of the wing such as tip vortices, engine and the fuselage interference effects etc. Thus for a full-fledged implementation of the adaptive airfoil concept, it is necessary to extend the optimization methodology to 3-dimensional wings. Here, with an addition of a third dimension, more number of design variables will be required to completely parameterize the wing.

The optimization objectives can also be extended to include other performance parameters such as actuator energy requirements which are based on the energy required to overcome inertia, material stiffness and the aerodynamic loads in order to

morph the shape. This would require fluid structure interaction analysis, a subject that has drawn a lot of attention in the past few years.

The optimization methodology can be performed on more number of points inside the flight envelope. This would ensure that a much wider area of the flight envelope is covered, thus offering more operational flexibility while keeping the overall drag to a minimum throughout the flight operation.

The results from this research show that significant drag reduction is possible using the adaptive airfoil concept. This however must be weighed against the energy consumed in adapting the wing, the increased cost and complexity, and the implications for safety. As the actuator technology needed to deform the wing improves, the price of fuel increases, and the need to reduce harmful emissions becomes more critical, the adaptive wing concept will become increasingly viable.

REFERENCES

- [1] Boeing. (2012, 13/05/2012). *Boeing Long-Term Market Forecast summary*. Available: http://www.boeing.com/commercial/cmo/forecast_summary.html
- [2] J. E. Hicken, "Efficient algorithms for future aircraft design: Contributions to aerodynamic shape optimization," Doctor of Philosophy Doctoral thesis, Graduate Department of Aerospace Science and Engineering, University of Toronto, Toronto, Canada, 2009.
- [3] Airbus. (2012). *Biomimicry*. Available: <http://www.airbus.com/innovation/eco-efficiency/design/biomimicry/>
- [4] D. Lentink, U. K. Müller, E. J. Stamhuis, R. de Kat, W. van Gestel, L. L. M. Veldhuis, P. Henningsson, A. Hedenström, J. J. Videler, and J. L. van Leeuwen, "How swifts control their glide performance with morphing wings," *Nature*, vol. 446, pp. 1082-1085.
- [5] T. A. Weisshaar, "Morphing Aircraft Technology - New Shapes for Aircraft Design," 01 OCT 2006 2012.
- [6] G. E. Lewis, R. E. Thomasson, and D. W. Nelson, "Wing Lift/Drag Optimizing System," 1990.
- [7] R. W. Wlezien, G. C. Homer, and A. R. McGowan, "THE AIRCRAFT MORPHING PROGRAM," American Institute of Aeronautics and Astronautics, Long Beach, California, USA1998.
- [8] G. Dockter, "Geometric Morphing Wing with Exapandable Spars," United States of America Patent, 2003.
- [9] M. K. V. Sankrithi and J. B. Frommer, "Controllable Winglets," 2010.
- [10] S. National Research Council . Division on Engineering and Physical and A. National Research Council . Steering Committee for the Decadal Survey of Civil, *Decadal survey of civil aeronautics: foundation for the future*. Washington, D.C: The National Academies Press, 2006.
- [11] E. Stanewsky, "Adaptive wing and flow control technology," *Progress in Aerospace Sciences*, vol. 37, pp. 583-667, 2001.
- [12] N. J. Pern, Jacob J. D., "Aerodynamic Flow Control Using Shape Adaptive Surfaces," in *ASME Design Engineering Technical Conferences*, Las Vegas, Nevada, USA, 1999.

- [13] A. Jameson, N. A. Pierce, and L. Martinelli, "Optimum Aerodynamic Design using the Navier-Stokes Equations," *AIAA*, 1997.
- [14] A. Jameson, R. Hu, and A. C. Brown, "Aerodynamic Shape Optimization for the World's Fastest P-51," presented at the 44th Aerospace Sciences Meeting and Exhibit, Reno, Nevada, 2006.
- [15] A. Jameson, "Aerodynamic Design via Control Theory," *Journal of Scientific Computing*, vol. 3, 1988.
- [16] A. Jameson, "Optimum Aerodynamic Design using CFD and Control Theory," *AIAA*, 1995.
- [17] F. Austin, M. J. Rossi, W. van Nostrand, J. Su, G. Knowles, and A. Jameson, "Active Rib Experiment for Adaptive Conformal Wing," in *Third International Conference on Adaptive Structures*, 00/1993, p. 43.
- [18] D. W. Zingg, L. Diosady, and L. Billing, "Adaptive Airfoils for Drag Reduction at Transonic Speeds," presented at the 24th Applied Aerodynamic Conference, San Francisco, California, 2006.
- [19] H. Namgoong, W. A. Crossley, and A. S. Lyrintzis, "Aerodynamic Optimization of a Morphing Airfoil Using Energy as an Objective," *AIAA journal*, vol. 45, pp. 2113-2124, 2007.
- [20] P. Gamboa, J. Vale, P. Lau, and A. Suleman, "Optimization of a Morphing Wing Based on Coupled Aerodynamic and Structural Constraints," *AIAA journal*, vol. 47, 2009.
- [21] N. M. Ursache, "The Design of Adaptive Structures for Wing Morphing," Ph.D, Faculty of Engineering, Science and Mathematics, University of Southampton, 2006.
- [22] L. Unlusoy, "Structural Design and Analysis of the Mission Adaptive Wings of an Unmanned Aerial Vehicle," Master's, Graduate School of Natural and Applied Sciences, Middle East Technical University, 2010.
- [23] J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed. New York, NY: Springer Science+Business Media, LLC, 2006.
- [24] S. S. Rao, *Engineering Optimization - Theory and Practice* 4ed.: John Wiley & Sons.
- [25] E. Üstündag and M. S. Çelebi, "A B-Spline Curve Fitting Approach by Implementing the Parameter Correction Terms."
- [26] J. Hoschek, F. Schneider, and P. Wassum, "Optimal Approximate Conversion of Spline Surfaces," *Computer-Aided Design* 6, pp. 293 -306, 1983.

- [27] L. K. Billing, "On The Development Of An Improved Lift-Constrained Aerodynamic Optimization Algorithm," Masters Thesis, Graduate Department of Aerospace Engineering, University of Toronto, Toronto, Canada, 2006.
- [28] J. D. Anderson, *Fundamentals of aerodynamics*. Boston, Mass: McGraw-Hill, 2001.
- [29] J. H. Ferziger and M. Peric, *Computational Methods for Fluid Dynamics*, 3rd ed.: Springer, 2002.
- [30] J. M. McDonough, "Introductory Lectures on Turbulence," ed. Kentucky, USA: University of Kentucky, 2004.
- [31] *ANSYS FLUENT Theory Guide Release 13.0*. USA: ANSYS, Inc., 2010.
- [32] M. El-Sayed, T. Sun, and J. Berry, "Shape optimization with computational fluid dynamics," *Advances in Engineering Software*, vol. 36, pp. 607-613, 2005.
- [33] A. I. J. Forrester and A. J. Keane, "Recent advances in surrogate-based optimization," *Progress in Aerospace Sciences*, vol. 45, pp. 50-79, 2009.
- [34] J. Ahn, H. J. Kim, D. H. Lee, and O. H. Rho, "Response surface method for airfoil design in transonic flow," *Journal of aircraft*, vol. 38, pp. 231-238, 2001.
- [35] V. N. e. a. Queipo, "Surrogate-based analysis and optimization," *Progress in Aerospace Sciences*, vol. 41, 2005.
- [36] Y. Mack and e. al, "Surrogate Model-Based Optimization Framework: A Case Study in Aerospace Design," *Studies in Computational Intelligence (SCI)*, vol. 51, pp. 323-342, 2007.
- [37] A. Forrester, A. Sobester, and A. Keane, *Engineering Design via Surrogate Modelling: A Practical Guide*: Wiley, 2008.
- [38] A. Vavalle and N. Qin, "Iterative response surface based optimization scheme for transonic airfoil design," *Journal of aircraft*, vol. 44, pp. 365-376, 2007.
- [39] *MATLAB Optimization Toolbox™ 5 User's Guide* Mathworks, 2010.
- [40] *MATLAB Global Optimization Toolbox-User's Guide* Mathworks, 2010.

Appendix A GridPro Topology Input Language (TIL) File (FINAL3.FRA)

```
SET DIMENSION 2
SET GRIDDED 16
SET DISPLAY.SURF ON
COMPONENT main()
BEGIN
INPUT 1 surf(sOUT (0..7));
INPUT 2 corn(sIN (1:1..8),cIN (-129));
END

COMPONENT surf()
BEGIN
s 0 -linear "GPro_af1.dat" ; #TIL:1:0
s 1 -linear "GPro_af2.dat" ; #TIL:1:1
s 2 -linear "C_L.dat" ; #TIL:1:2
s 3 -linear "Back_end.dat" ; #TIL:1:3
s 4 -linear "Wake_line.dat" -O ; #TIL:1:4
s 5 -linear "trail_top.dat" -O ; #TIL:1:5
s 6 -linear "trail_down.dat" -O ; #TIL:1:6
s 7 -linear "af_lead.dat" -O ; #TIL:1:7
LABEL UnknownLabel_0= s(2);
LABEL UnknownLabel_1= s(3);
LABEL UnknownLabel_2= s(2);
LABEL UnknownLabel_3= s(3);
LABEL UnknownLabel_4= s(2);
LABEL UnknownLabel_5= s(3);
LABEL UnknownLabel_6= s(2);
LABEL UnknownLabel_7= s(3);
LABEL UnknownLabel_8= s(2);
LABEL UnknownLabel_9= s(3);
LABEL UnknownLabel_10= s(2);
LABEL UnknownLabel_11= s(3);
LABEL UnknownLabel_12= s(2);
LABEL UnknownLabel_13= s(3);
LABEL UnknownLabel_14= s(2);
```



```

LABEL UnknownLabel_15= s(3);
LABEL UnknownLabel_16= s(2);
LABEL UnknownLabel_17= s(3);
LABEL _009_user9= s(2);
LABEL _009_user9= s(3);
END

```

```

COMPONENT corn(sIN s[0..7],cIN c[0..128])

```

```

BEGIN

```

```

c 0 15 -20 0 -s s:3 s:2 -L c:0 -p 15.92 -15.27 0 ; #TIL:2:0
c 1 15 20 0 -s s:3 s:2 -L c:1 -p 15.55 16.25 0 ; #TIL:2:1
c 2 -15.379608 12.026067 0 -s s:2 -L c:2 -p -4.856 16.29 0 -g 1;
#TIL:2:2
c 3 -14.208652 -13.075139 0 -s s:2 -L c:3 -p -
16.01 -4.648 0 -g 1; #TIL:2:3
c 4 0.27747889 0.06994594 0 -s s:0 -L c:4 -p -4.754
5.103 0 -g 2033; #TIL:2:4
c 5 0.75113867 0.03877387 0 -s s:0 -L c:5 -p 5.094 5.147
0 -g 2033; #TIL:2:5
c 6 0.29804802 -0.05459776 0 -s s:1 -L c:6 -p -
4.666 -4.536 0 ; #TIL:2:6
c 7 0.70580358 -0.03767442 0 -s s:1 -L c:7 -p 5.193 -
4.439 0 ; #TIL:2:7
c 8 -0.53387156 1.1149678 0 -L c:8 -p -4.754 5.103 0 -g
1; #TIL:2:8
c 9 -0.49950319 -1.0935291 0 -L c:9 -p -4.666 -4.536 0
-g 1; #TIL:2:9
c 10 -18.427811 -5.2222994 0 -s s:2 -L c:10 -p -16.05
0.3627 0 -g 1; #TIL:2:10
c 11 -0.97522092 -0.46675736 0 -L c:11 -p -4.711
0.4427 0 -g 1; #TIL:2:11
c 12 0.01205125 -0.02173760 0 -s s:1 -L c:12 -p -
4.711 0.4427 0 -g 33; #TIL:2:12
c 13 0.10163954 -0.04295817 0 -s s:1 -L c:13 12 6 -p -
4.688 -2.158 0 -g 1;#TIL:2:13
g 13 12 40 ;
g 13 6 54 ;
c 14 -0.78115013 -0.78870751 0 -L c:14 11 9 -p -
4.688 -2.158 0 -g 1;#TIL:2:14
c 15 -16.756227 -9.3594344 0 -s s:2 -L c:15 10 3 -p -16.03
-2.255 0 -g 1;#TIL:2:15

```

c 16 0.09806832 0.05570585 0 -s s:0 -L c:16 4 -p -4.733
2.837 0 -g 2033;#TIL:2:16
g 16 4 54 ;
c 17 -0.81260331 0.7956845 0 -L c:17 8 -p -4.733 2.837 0 -
g 1;#TIL:2:17
c 18 -17.386961 8.2344006 0 -s s:2 -L c:18 2 -p -
16.06 2.772 0 -g 1;#TIL:2:18
c 19 0.00715576 0.03070321 0 -s s:0 -L c:19 16 -p -4.719
1.323 0 -g 2033;#TIL:2:19
g 19 16 40 ;
c 20 -0.98964462 0.46095311 0 -L c:20 17 -p -4.719 1.323 0 -
g 1;#TIL:2:20
c 21 -18.647731 4.6642091 0 -s s:2 -L c:21 18 -p -16.05
1.249 0 -g 1; #TIL:2:21
c 22 1.0000000000 0.0000000000 0 -s s:6 s:5 s:4 s:1 s:0 -L
c:22 5 7 -p 5.133 1.325 0 -g 2041;#TIL:2:22
g 22 5 60 ;
g 22 7 60 ;
c 23 15 -0.0983066870 0 -s s:4 s:3 -L c:23 -p 15.72 1.395
0 -g 17; #TIL:2:23
c 24 0.4849936670 -0.0512852062 0 -s s:1 -L c:24 7 6 -p
1.077 -4.479 0 -g 1;#TIL:2:24
g 24 7 54 ;
g 24 6 54 ;
c 25 -0.15831072 -1.3593258 0 -L c:25 9 -p 1.077 -
4.479 0 -g 1;#TIL:2:25
c 26 -10.54709 -16.671646 0 -s s:2 -L c:26 3 -p 1.185 -
15.51 0 -g 1; #TIL:2:26
c 27 0.4931796914 0.0639134891 0 -s s:0 -L c:27 5 4 -p 0.1027
5.125 0 -g 2033;#TIL:2:27
g 27 5 54 ;
g 27 4 54 ;
c 28 -0.17851629 1.3853468 0 -L c:28 8 -p 0.1027
5.125 0 -g 1;#TIL:2:28
c 29 -11.830347 15.728489 0 -s s:2 -L c:29 2 -p -0.006178
16.28 0 -g 1; #TIL:2:29
c 30 0.0042123225 0.0347450144 0 -L c:30 19 -p -4.719
1.323 0 -g 1; #TIL:2:30
g 30 19 24 ;
c 31 0.0971142250 0.0606139721 0 -L c:31 16 30 -p -4.733 2.837
0 -g 1; #TIL:2:31

c 32 0.2773615447 0.0749445585 0 -L c:32 4 31 -p -
4.754 5.103 0 -g 1; #TIL:2:32
c 33 0.4935498630 0.0688997675 0 -L c:33 27 32 -p
0.1027 5.125 0 -g 1; #TIL:2:33
c 34 0.7517889821 0.0437313968 0 -L c:34 5 33 -p 5.094
5.147 0 -g 1; #TIL:2:34
c 35 0.7062760733 -0.0426520439 0 -L c:35 7 -p 5.193 -
4.439 0 -g 1; #TIL:2:35
c 36 0.4852184837 -0.0562801494 0 -L c:36 24 35 -p 1.077
-4.479 0 -g 1; #TIL:2:36
c 37 0.2979851125 -0.0595973593 0 -L c:37 6 36 -p -
4.666 -4.536 0 -g 1; #TIL:2:37
c 38 0.1009855635 -0.0479152191 0 -L c:38 13 37 -p -
4.688 -2.158 0 -g 1; #TIL:2:38
c 39 0.0095806800 -0.0260845820 0 -L c:39 12 38 -p -
4.711 0.4427 0 -g 1;#TIL:2:39
c 40 1 20 0 -s s:5 s:2 -L c:40 -p 5.331 16.27
0 -g 1;#TIL:2:40
c 41 1 -20 0 -s s:6 s:2 -L c:41 -p 6.534 -
15.42 0 -g 1; #TIL:2:41
c 42 -0.003 0 0 -s s:7 s:1 s:0 -L c:42 19 12 -p -4.715
0.8951 0 -g 33; #TIL:2:42
g 42 19 36 ;
g 42 12 36 ;
c 43 -0.008 0 0 -s s:7 -L c:43 30 39 42 -p -4.715 0.8951 0
-g 1;#TIL:2:43
c 44 -1.07 0 0 -s s:7 -L c:44 20 11 -p -4.715 0.8951 0
-g 1;#TIL:2:44
c 45 -19.1 0 0 -s s:7 s:2 -L c:45 21 10 -p -16.05
0.8182 0 -g 1;#TIL:2:45
c 46 0.8220226547 0.5791445375 0 -L c:46 34 -p 5.094 5.147
0 -g 1;#TIL:2:46
g 46 34 18 ;
c 47 0.7573058268 -0.5802354926 0 -L c:47 35 -p 5.193 -4.439 0 -
g 1;#TIL:2:47
c 48 0.5032038178 -0.4558756045 0 -L c:48 36 25 47 -p 1.077
-4.479 0 -g 1;#TIL:2:48
g 48 25 8 ;
c 49 0.2942105293 -0.3595736126 0 -L c:49 37 9 48 -p -
4.666 -4.536 0 -g 1;#TIL:2:49
c 50 0.0695948522 -0.2858534952 0 -L c:50 38 14 49 -p -
4.688 -2.158 0 -g 1;#TIL:2:50

c 51 -0.0892421031 -0.1999640159 0 -L c:51 39 11 50 -p -4.711
0.4427 0 -g 1;#TIL:2:51
c 52 -0.208 0 0 -s s:7 -L c:52 43 44 51 -p -4.715
0.8951 0 -g 1;#TIL:2:52
c 53 -0.1135251564 0.1964171705 0 -L c:53 30 20 52 -p -
4.719 1.323 0 -g 1;#TIL:2:53
c 54 0.0513175085 0.2962039990 0 -L c:54 31 17 53 -p -4.733
2.837 0 -g 1;#TIL:2:54
c 55 0.2703208312 0.3748619277 0 -L c:55 32 8 54 -p -
4.754 5.103 0 -g 1;#TIL:2:55
c 56 0.5231635916 0.4678020451 0 -L c:56 33 28 55 46 -p
0.1027 5.125 0 -g 1;#TIL:2:56
c 57 15 4.75 0 -s s:3 -L c:57 -p 15.7 2.774 0 -g
1;#TIL:2:57
c 58 15 -4.75 0 -s s:3 -L c:58 -p 15.74 -0.1492 0 -g
1;#TIL:2:58
c 59 1 1.9 0 -s s:5 -L c:59 -p 5.142 2.025 0 -g
12582913;#TIL:2:59
c 60 1 -1.9 0 -s s:6 -L c:60 -p 5.191 0.6298 0 -g 1;
#TIL:2:60
c 61 1 0.05 0 -s s:5 -L c:61 22 34 -p 5.133 1.342 0 -g
1;#TIL:2:61
c 62 15 0.9016933130 0 -s s:3 -L c:62 23 -p 15.72 1.429 0 -g
1;#TIL:2:62
c 63 1 -0.05 0 -s s:6 -L c:63 22 35 -p 5.135 1.303 0 -g
1;#TIL:2:63
c 64 15 -1.0983066870 0 -s s:3 -L c:64 23 -p 15.72 1.346
0 -g 1;#TIL:2:64
c 65 0.30973 1.6663 0 -L c:65 59 28 46 -p 2.868 3.424
0 -g 1;#TIL:2:65
c 66 -6.6558587 18.803912 0 -s s:2 -L c:66 40 29 -p
2.923 16.27 0 -g 1;#TIL:2:66
c 67 1 0.6500000000 0 -s s:5 -L c:67 61 59
46 -p 5.136 1.582 0 -g 1;#TIL:2:67
c 68 15 2.0116933130 0 -s s:3 -L c:68 62 57
-p 15.71 1.901 0 -g 1;#TIL:2:68
c 69 1 -0.6500000000 0 -s s:6 -L c:69 63 60 47 -p 5.156
1.047 0 -g 1;#TIL:2:69
c 70 15 -2.2083066870 0 -s s:3 -L c:70 64 58 -p 15.73
0.7777 0 -g 1;#TIL:2:70
c 71 0.32361616 -1.6274478 0 -L c:71 60 25 47 -p
3.62 -1.321 0 -g 1;#TIL:2:71
c 72 -5.1198025 -19.27377 0 -s s:2 -L c:72 41 26 -p
4.492 -15.45 0 -g 1;#TIL:2:72

c 73 1 4.3299106 0 -s s:5 -L c:73 59 -p
5.175 4.507 0 -g 98305;#TIL:2:73
g 73 59 8 ;
c 74 15 7 0 -s s:3 -L c:74 57 -p 15.67 5.122
0 -g 12289;#TIL:2:74
c 75 -0.58536495 3.8241013 0 -L c:75 65 73 -p
2.878 5.662 0 -g 98305;#TIL:2:75
c 76 -1.4907674 3.3199326 0 -L c:76 28 75 -p
0.08373 7.069 0 -g 1;#TIL:2:76
c 77 -2.3056719 2.6722406 0 -L c:77 8 76 -p -
4.772 7.052 0 -g 1;#TIL:2:77
c 78 -2.9337173 1.9277671 0 -L c:78 17 77 -p -
6.706 2.826 0 -g 393217;#TIL:2:78
c 79 -3.3924358 0.97611577 0 -L c:79 20 78 -p
-6.693 1.31 0 -g 393217;#TIL:2:79
c 80 -3.57757 0 0 -s s:7 -L c:80 44 79
-p -6.69 0.8817 0 -g 393217;#TIL:2:80
c 81 -3.3839423 -1.1899742 0 -L c:81 11 80 -p -6.687
0.4288 0 -g 393217;#TIL:2:81
c 82 -2.9437786 -1.9421683 0 -L c:82 14 81 -p
-6.664 -2.175 0 -g 393217;#TIL:2:82
c 83 -2.3087376 -2.6727836 0 -L c:83 9 82 -p
-6.642 -4.556 0 -g 1;#TIL:2:83
c 84 -1.5067772 -3.2887937 0 -L c:84 25 83 -p
1.096 -6.401 0 -g 1;#TIL:2:84
c 85 -0.36163206 -3.884064 0 -L c:85 71 84
-p 3.772 -3.783 0 -g 1;#TIL:2:85
c 86 1 -4.2415517 0 -s s:6 -L c:86 60 85 -p 5.425
-2.167 0 -g 1572865;#TIL:2:86
c 87 15 -7 0 -s s:3 -L c:87 58 -p
15.77 -2.784 0 -g 1572865;#TIL:2:87
c 88 1 9.3 0 -s s:5 -L c:88 40 73 -p 5.276
12.11 0 -g 32769;#TIL:2:88
g 88 40 8 ;
g 88 73 8 ;
c 89 15 11.5 0 -s s:3 -L c:89 1 74
-p 15.59 12.31 0 -g 4097;#TIL:2:89
c 90 -2.1768669 8.5100887 0 -L c:90 66 88 75 -p 2.907
12.52 0 -g 32769;#TIL:2:90
c 91 -4.3865592 7.5781182 0 -L c:91 29 90 76 -p
0.02562 13.02 0 -g 1;#TIL:2:91
c 92 -6.2301001 6.0427238 0 -L c:92 2 91 77 -p
-4.826 13.02 0 -g 1;#TIL:2:92

c 93 -7.5858052 4.2258478 0 -L c:93 18 92 78 -p
-12.75 2.791 0 -g 131073;#TIL:2:93
c 94 -8.421654 2.1496 0 -L c:94 21 93 79 -p
-12.74 1.271 0 -g 131073;#TIL:2:94
c 95 -8.65483 0 0 -s s:7 -L c:95 45 94 80
-p -12.74 0.8407 0 -g 131073;#TIL:2:95
c 96 -8.344819 -2.3600329 0 -L c:96 10 95 81 -p -
12.74 0.3861 0 -g 131073;#TIL:2:96
c 97 -7.6475522 -4.1163208 0 -L c:97 15 96 82 -p
-12.72 -2.227 0 -g 131073;#TIL:2:97
c 98 -6.3737558 -5.9080515 0 -L c:98 3 97 83 -p
-12.7 -4.615 0 -g 1;#TIL:2:98
c 99 -4.301992 -7.6013844 0 -L c:99 26 98 84 -p 1.153
-12.29 0 -g 1;#TIL:2:99
c 100 -1.7742141 -8.7165805 0 -L c:100 72 99 85
-p 4.237 -11.32 0 -g 1;#TIL:2:100
c 101 1 -9.3 0 -s s:6 -L c:101 41 100
86 -p 6.142 -10.73 0 -g 2621441;#TIL:2:101
c 102 15 -11.5 0 -s s:3 -L c:102 0 87 -p
15.87 -10.85 0 -g 2621441;#TIL:2:102
c 103 6.3 20 0 -s s:2 -L c:103 40 -p
8.677 16.26 0 -g 1;#TIL:2:103
g 103 40 90 ;
c 104 6.3 10.2 0 -L c:104 88 103 -p 8.654
12.18 0 -g 4097;#TIL:2:104
c 105 6.20049 5.6128913 0 -L c:105 73 104 -p
8.612 4.708 0 -g 12289;#TIL:2:105
c 106 6.0137047 3.5208954 0 -L c:106 59 105 -p
8.598 2.27 0 -g 50331649;#TIL:2:106
c 107 5.74 1.2384014220 0 -L c:107 106 67 -p 8.598
1.686 0 -g 1;#TIL:2:107
c 108 5.6 0.5184014220 0 -L c:108 107 61 -p 8.599
1.37 0 -g 1;#TIL:2:108
c 109 5.5 -0.0315985780 0 -s s:4 -L c:109 108 22 -p
8.599 1.348 0 -g 17;#TIL:2:109
c 110 5.6 -0.5815985780 0 -L c:110 109 63 -p 8.6
1.317 0 -g 1;#TIL:2:110
c 111 5.74 -1.3015985780 0 -L c:111 110 69 -p
8.618 0.9588 0 -g 1;#TIL:2:111
c 112 5.9140989 -3.2509713 0 -L c:112 60 111 -p
8.645 0.3748 0 -g 1;#TIL:2:112
c 113 6.1106403 -5.5190698 0 -L c:113 86 112 -p
8.812 -2.369 0 -g 1572865;#TIL:2:113

```

c 114 6.3 -10.2 0 -L c:114 101 113 -p 9.326
-10.77 0 -g 2621441;#TIL:2:114
c 115 6.3 -20 0 -s s:2 -L c:115 41 114
-p 9.607 -15.37 0 -g 1;#TIL:2:115
c 116 11.2 20 0 -s s:2 -L c:116 103 1 -p
12.14 16.26 0 -g 1;#TIL:2:116
g 116 103 30 ;
g 116 1 30 ;
c 117 11.2 10.946 0 -L c:117 104 89 116
-p 12.15 12.25 0 -g 4097;#TIL:2:117
c 118 11.083306 6.5259738 0 -L c:118 105 74 117 -p 12.17
4.917 0 -g 12289;#TIL:2:118
c 119 10.976038 4.4195581 0 -L c:119 106 57 118 -p
12.18 2.524 0 -g 1;#TIL:2:119
c 120 10.865314 1.4825897003 0 -L c:120 107 68
119 -p 12.19 1.795 0 -g 1;#TIL:2:120
c 121 10.7 0.6825897003 0 -L c:121 108 62 120 -p 12.19 1.4 0
-g 1;#TIL:2:121
c 122 10.6 -0.0674102997 0 -s s:4 -L c:122 109 23 121
-p 12.19 1.372 0 -g 17;#TIL:2:122
c 123 10.7 -0.8174102997 0 -L c:123 110 64 122 -p
12.19 1.332 0 -g 1;#TIL:2:123
c 124 10.819722 -1.6174102997 0 -L c:124 111 70
123 -p 12.21 0.8675 0 -g 1;#TIL:2:124
c 125 10.935462 -4.0232149 0 -L c:125 112 58 124
-p 12.22 0.1105 0 -g 1;#TIL:2:125
c 126 10.989641 -6.2251315 0 -L c:126 113 87 125
-p 12.32 -2.578 0 -g 1572865;#TIL:2:126
c 127 11.2 -10.8775 0 -L c:127 114 102 126 -p
12.63 -10.81 0 -g 2621441;#TIL:2:127
c 128 11.2 -20 0 -s s:2 -L c:128 115
0 127 -p 12.79 -15.32 0 -g 1;#TIL:2:128
END

```

Schedule File (TIL)

File (FINAL3.SCH)

```

step 1: -S 1000 -w
write -f FINAL3.tmp

```

MATLAB Run Commands

```

>> !Ggrid FINAL3.fra
>> !clu FINAL3.tmp -s 1 0.00025 1.05

```

```
>> movefile('FINAL3.tmp.conn','FINAL3.tmp.tmp.conn');
>>! "C:\GridPro\az_mngr\outU_fluent.script.bat"
"C:/~/~/FINAL3.tmp.tmp" "C:/~/~/FLO_MESH.msh"
```

Appendix B ANSYS-FLUENT JOURNAL FILE (FLUENT_RUN.JOU)

```
(cx-gui-do cx-activate-item "MenuBar*ReadSubMenu*Mesh...")
(cx-gui-do cx-set-text-entry "Select File*FilterText" "D:\MASTERS
THESIS\CFD work\F I N A L - O P E R A T I O N\CFD SLAVE\*")
(cx-gui-do cx-activate-item "Select File*Apply")
(cx-gui-do cx-set-text-entry "Select File*Text" "FLO_MESH.msh")
(cx-gui-do cx-activate-item "Select File*OK")
/define/models/viscous/spalart-allmaras? yes
/define/materials/change-create air air yes ideal-gas no no yes
sutherland three-coefficient-method 1.716e-05 273.11 110.56 no no no
/mesh/modify-zones/merge-zones 10 11 ()
/mesh/modify-zones/merge-zones 2 3 4 5 6 7 ()
(load "OP.var")
/define/boundary-conditions/pressure-far-field pressure-far-field-10
no 0 no M no T no M_x no M_y no n yes no 10
/define/operating-conditions/operating-pressure P
/report/reference-values/compute pressure-far-field pressure-far-
field-10
/solve/set/p-v-coupling 24
(cx-gui-do cx-activate-item
"NavigationPane*Frame1*PushButton12(Solution Methods)")
/solve/set/discretization-scheme/density 1
/solve/set/discretization-scheme/pressure 12
/solve/set/discretization-scheme/mom 1
/solve/set/discretization-scheme/nut 1
/solve/set/discretization-scheme/temperature 1
/solve/set/under-relaxation/nut 0.9
(cx-gui-do cx-activate-item
"NavigationPane*Frame1*PushButton13(Solution Controls)")
(cx-gui-do cx-set-real-entry-list "Solution
Controls*Frame1*Table1*Frame4(Explicit Relaxation
Factors)*Table4(Explicit Relaxation Factors)*RealEntry1(Momentum)" '(
0.5))
```



```

(cx-gui-do cx-activate-item "Solution
Controls*Frame1*Table1*Frame4(Explicit Relaxation
Factors)*Table4(Explicit Relaxation Factors)*RealEntry1(Momentum)")

(cx-gui-do cx-set-real-entry-list "Solution
Controls*Frame1*Table1*Frame4(Explicit Relaxation
Factors)*Table4(Explicit Relaxation Factors)*RealEntry3(Pressure)" '(
0.5))

(cx-gui-do cx-activate-item "Solution
Controls*Frame1*Table1*Frame4(Explicit Relaxation
Factors)*Table4(Explicit Relaxation Factors)*RealEntry3(Pressure)")

/solve/monitors/residual/convergence-criteria 0.0000001 0.0001 0.0001
1e-06 0.001

/solve/monitors/force/drag-coefficient yes 8 9 ( ) yes no yes 2 no D_x
D_y

/solve/monitors/force/lift-coefficient yes 8 9 ( ) yes no yes 3 no L_x
L_y

/solve/monitors/force/moment-coefficient yes 8 9 ( ) yes no yes 4 no
0.25 0 0 0 1

/solve/initialize/hyb-initialization

/solve/iterate 300

/report/forces/wall-forces yes D_x D_y yes "DRAG.csv"

/report/forces/wall-forces yes L_x L_y yes "LIFT.csv"

/report/forces/wall-moments yes 0.25 0 0 0 1 yes "MOMENT.csv"

/plot/plot yes "CP_TOP.csv" no no no pressure-coefficient y 1 0 wall-
8 ( )

/plot/plot yes "CP_LOW.csv" no no no pressure-coefficient y 1 0 wall-
9 ( )

/plot/plot yes "Mach_TOP.csv" no no no mach-number y 1 0 wall-8 ( )

/plot/plot yes "Mach_LOW.csv" no no no mach-number y 1 0 wall-9 ( )

/plot/plot yes "Stress_TOP.csv" no no no wall-shear y 1 0 wall-8 ( )

/plot/plot yes "Stress_LOW.csv" no no no wall-shear y 1 0 wall-9 ( )

exit

ok

```

VITA

Mohammed Taha Shafiq Khot was born on September 4, 1987, in the emirate of Dubai, United Arab Emirates. He is a citizen of the republic of India. He attended the Sharjah Indian School where he studied till grade 10. He then moved to India where he attended high school at R. J. College, Mumbai in 2005. After high school, he joined University of Mumbai—K. J. Somaiya College of Engineering where he studied Mechanical Engineering, graduating with a Bachelor's degree (in First Class) in September 2009.

After graduating, Mr. Taha joined the American University of Sharjah where he pursued a master's program in Mechanical Engineering. He was awarded the Master of Science in Mechanical Engineering degree in 2012.