

Washington University in St. Louis
Washington University Open Scholarship

Engineering and Applied Science Theses &
Dissertations

McKelvey School of Engineering

Summer 8-2014

Shape Optimization of Airfoils Without and With Ground Effect Using a Multi-Objective Genetic Algorithm

Yilei He

Washington University in St Louis

Follow this and additional works at: https://openscholarship.wustl.edu/eng_etds



Part of the [Mechanical Engineering Commons](#)

Recommended Citation

He, Yilei, "Shape Optimization of Airfoils Without and With Ground Effect Using a Multi-Objective Genetic Algorithm" (2014).
Engineering and Applied Science Theses & Dissertations. 5.
https://openscholarship.wustl.edu/eng_etds/5

This Thesis is brought to you for free and open access by the McKelvey School of Engineering at Washington University Open Scholarship. It has been accepted for inclusion in Engineering and Applied Science Theses & Dissertations by an authorized administrator of Washington University Open Scholarship. For more information, please contact digital@wumail.wustl.edu.

WASHINGTON UNIVERSITY IN ST. LOUIS
School of Engineering and Applied Science
Department of Mechanical Engineering and Material Science

Thesis Examination Committee:
Ramesh Agarwal (Chair)
David Peters
Kenneth Jerina

SHAPE OPTIMIZATION OF AIRFOILS WITHOUT AND WITH GROUND EFFECT
USING A MULTI-OBJECTIVE GENETIC ALGORITHM

by
Yilei He

A thesis presented to the School of Engineering and Applied Science
of Washington University in partial fulfillment of the
requirements for the degree of
Master of Science

August 2014

Saint Louis, Missouri

Contents

List of Figures.....	iii
List of Tables.....	vii
List of Abbreviations.....	viii
Acknowledgments.....	ix
Abstract.....	xi
1 Introduction.....	1
1.1 Motivation.....	1
1.2 Brief Review of Literature.....	2
1.3 Scope of the Thesis.....	3
2 Optimization Methodology and Flow Field Simulations of Original S809 and NACA 4412 Airfoils.....	4
2.1 Genetic Algorithm based Optimization Method.....	5
2.1.1 Overview of genetic algorithm.....	5
2.1.2 Multi-objective Genetic Algorithm (MOGA).....	6
2.1.3 Implementation of multi-objective genetic algorithm (MOGA).....	7
2.2 Airfoil Parameterization.....	7
2.3 Mesh Generation.....	9
2.4 Flow Field Computations on Original S809 Airfoil and Tyrrell-026 Airfoil in Ground Effect for Code Validation.....	11
2.4.1 Wind-Turbine S809 Airfoil.....	11
2.4.2 Tyrrell-026 Airfoil in Ground Effect.....	14
3 Optimization Results.....	15
3.1 Results for optimized S809 wind turbine airfoil.....	15
3.2 Results for optimized NACA 4412 airfoil.....	31
4 Conclusions.....	50
References.....	51
Appendix.....	53
A.1 NSGAII_main.java.....	53
A.2 MOGA.java.....	54
A.3 AirfoilNRELS809.java.....	57

A.4	AirfoilNRELS809.java	60
A.5	ReadRecord.java.....	66
A.6	WriteRecord.java.....	73
A.7	NRELS809.Jou.....	75
A.8	ICEMS809.rpl	83
Vita	93

List of Figures

Figure 1: Schematic of information flow in the optimization process	4
Figure 2: Illustration of the general Crossover function in GA	5
Figure 3: Distribution of Bezier-curve control points on S809 airfoil	9
Figure 4: Structured C-grid mesh around the NREL S809 airfoil	10
Figure 5: Structured Mesh around the NACA 4412 airfoil in ground effect.....	10
Figure 6: Variation in lift coefficient with angle of attack	13
Figure 7: Variation in drag coefficient with angle of attack	13
Figure 8: Comparison of computed and experimental pressure coefficient for the Tyrrell-026 airfoil.....	14
Figure 9: Velocities in the rotor plan	15
Figure 10: Shape evolution of S809 airfoil at 23% span location at various generations of MOGA	18
Figure 11: Shape evolution of S809 airfoil at 58% span location at various generations of MOGA	19
Figure 12: Shape evolution of S809 airfoil at 98% span location at various generations of MOGA	20
Figure 13: Pareto front for S809 airfoils at 23% span location.....	21
Figure 14: Pareto front for S809 airfoils at 58% span location.....	21
Figure 15: Pareto front for S809 airfoils at 98% span location.....	22
Figure 16(a): Comparison of shape of optimized S809 airfoil and original S809 airfoil at 23% span location	22
Figure 16(b): Comparison of pressure coefficient of optimized S809 airfoil and the original S809 airfoil at 23% span location	23
Figure 17(a): Comparison of shape of optimized S809 airfoil and the original S809 airfoil at 58% span location.....	23
Figure 17(b): Comparison of pressure coefficient of optimized S809 airfoil and the original S809 airfoil at 58% span location	24
Figure 18(a): Comparison of shape of optimized S809 airfoil and the original S809 airfoil at 98% span location.....	24
Figure 18(b): Comparison of pressure coefficient of optimized S809 airfoil and the original S809 airfoil at 98% span location	25
Figure 19: Comparison of MOGA optimized S809 airfoil shape at 23% span location with that obtained by the adjoin method	25

Figure 20: Comparison of MOGA optimized S809 airfoil shape at 58% span location with that obtained by the adjoin method	26
Figure 21: Comparison of MOGA optimized S809 airfoil shape at 98% span location with that obtained by the adjoin method	26
Figure 22(a): Pressure contours around optimized S809 airfoil at 23% span location	27
Figure 22(b): Velocity contours around optimized S809 airfoil at 23% span location	27
Figure 23(a): Pressure contours around optimized S809 airfoil at 58% span location	28
Figure 23(b): Velocity contours around optimized S809 airfoil at 58% span location	28
Figure 24(a): Pressure contours around optimized S809 airfoil at 98% span location	29
Figure 24(b): Velocity contours around optimized S809 airfoil at 98% span location	29
Figure 25: Comparison of shape between the optimal airfoil with 0.2C ground clearance and the original airfoil	32
Figure 26(a): Variation of lift coefficient for the optimal airfoil with 0.2C ground clearance and the original airfoil for different ground heights	33
Figure 26(b): Variation of lift to drag ratio for the optimal airfoil with 0.2C ground clearance and the original airfoil for different ground heights	33
Figure 27(a): Comparison of pressure coefficient between the optimal airfoil with 0.2C ground clearance and original airfoil.....	34
Figure 27(b): Pressure contours around the optimal airfoil with 0.2C ground clearance	34
Figure 27(c): Velocity contours around the optimal airfoil with 0.2C ground clearance.....	35
Figure 28(a): Comparison of pressure coefficient between the optimal airfoil with ground clearance of 0.2C and the original airfoil at ground height of 0.4C	35
Figure 28(b): Pressure contours around the airfoil optimized for ground clearance of 0.2C at ground height of 0.4C	36
Figure 28(c): Velocity contour around the airfoil optimized for ground clearance of 0.2C at ground height of 0.4C	36
Figure 29(a) : Comparison of pressure coefficient between the optimal airfoil with ground clearance of 0.2C and the original airfoil at ground height of 0.6C	37
Figure 29(b): Pressure contours around the airfoil optimized for ground clearance of 0.2C at ground height of 0.6C	37
Figure 29(c): Velocity contour around the airfoil optimized for ground clearance of 0.2C at ground height of 0.6C	38

Figure 30(a) : Comparison of pressure coefficient between the optimal airfoil with ground clearance of 0.2C and the original airfoil at ground height of 0.8C	38
Figure 30(b): Pressure contours around the airfoil optimized for ground clearance of 0.2C at ground height of 0.8C	39
Figure 30(c): Velocity contour around the airfoil optimized for ground clearance of 0.2C at ground height of 0.8C	39
Figure 31: Comparison of shape of airfoil optimized under free stream condition (without ground effect) and the original NACA 4412 airfoil	40
Figure 32: Comparison of shape of airfoil optimized under free stream condition (without ground effect) and the airfoil optimized in ground effect with 0.2C height above the ground	40
Figure 33: Comparison of lift to drag ration between the airfoil optimized in free stream (without ground effect), the airfoil optimized in ground effect with 0.2C height above the ground and the original NACA 4412 airfoil	41
Figure 34(a): Comparison of pressure coefficient of an airfoil optimized in free stream without ground effect and the original NACA 4412 airfoil at a ground height of 0.2C	42
Figure 34(b): Comparison of pressure coefficient of an airfoil optimized in free stream without ground effect and the airfoil optimized in ground effect with 0.2C height above the ground	42
Figure 34(c): Pressure contours around the airfoil optimized in free stream without ground effect at a ground height of 0.2C	43
Figure 34(d): Velocity contours around the airfoil optimized in free stream without ground effect at a ground height of 0.2C	43
Figure 35(a): Comparison of pressure coefficient of an airfoil optimized in free stream without ground effect and the original NACA 4412 airfoil at a ground height of 0.4C	44
Figure 35(b): Comparison of pressure coefficient of an airfoil optimized in free stream without ground effect and the airfoil optimized in ground effect with 0.4C height above the ground	44
Figure 35(c): Pressure contours around the airfoil optimized in free stream without ground effect at a ground height of 0.4C	45

Figure 35(d): Velocity contours around the airfoil optimized in free stream without ground effect at a ground height of $0.4C$	45
Figure 36(a): Comparison of pressure coefficient of an airfoil optimized in free stream without ground effect and the original NACA 4412 airfoil at a ground height of $0.6C$	46
Figure 36(b): Comparison of pressure coefficient of an airfoil optimized in free stream without ground effect and the airfoil optimized in ground effect with $0.6C$ height above the ground	46
Figure 36(c): Pressure contours around the airfoil optimized in free stream without ground effect at a ground height of $0.6C$	47
Figure 36(d): Velocity contours around the airfoil optimized in free stream without ground effect at a ground height of $0.6C$	47
Figure 37(a): Comparison of pressure coefficient of an airfoil optimized in free stream without ground effect and the original NACA 4412 airfoil at a ground height of $0.8C$	48
Figure 37(b): Comparison of pressure coefficient of an airfoil optimized in free stream without ground effect and the airfoil optimized in ground effect with $0.8C$ height above the ground	48
Figure 37(c): Pressure contours around the airfoil optimized in free stream without ground effect at a ground height of $0.8C$	49
Figure 37(d): Velocity contours around the airfoil optimized in free stream without ground effect at a ground height of $0.8C$	49

List of Tables

Table 1: Coordinates of the Bezier curve control points for S809 airfoil parameterization	8
Table 2: Coordinates of the Bezier curve control points for NACA 4412 airfoil parameterization	8
Table 3: Computed lift coefficients for S809 airfoil and their comparison with experimental lift coefficients	12
Table 4: Computed drag coefficients for S809 airfoil and their comparison with experimental drag coefficients	12
Table 5: Effective angle of attack and relative velocity experienced by the wind turbine airfoil for free stream wind velocity $V_\infty = 6.7\text{m/s}$, rotational speed $\omega = 72\text{ rpm}$ and pitch setting of 5 deg.....	16
Table 6: Comparison of Cl of optimized airfoils: MOGA results and those obtained by the Adjoint Method.....	30
Table 7: Comparison of Cl/Cd of optimized airfoils: MOGA results and those obtained by the Adjoint Method.....	30
Table 8: Lift to drag ratio (L/D) for different airfoils	41

List of Abbreviations

Cl: Lift Coefficient

Cd: Drag Coefficient

Cp: Pressure Coefficient

AoA: Angle of Attack

L/D: Lift to Drag Ratio

D/L: Drag to Lift Ratio

Acknowledgments

I would like to thank everyone who has helped me in completing this thesis providing technical help and encouragement. Thanks to Dr. Qu, who provided enormous help in answering many questions related to fluid dynamics and the intricacies of CFD simulations. Thanks to Dr. Cui for his help in preparation of my presentation slides. Thanks to Dr. Zhang and Dr. Chen for their continuous help and motivation.

Thanks to my committee members, Dr. Peters and Dr. Jerina, for taking the time to read the thesis and attend its defense.

Finally, a great amount of gratitude goes to my advisor Dr. Agarwal. I have learned a lot about CFD and other engineering knowledge from him. More importantly, he taught me how to do research and how to address a certain problem, this advice will benefit me forever in my career.

Yilei He

Washington University in St. Louis

August 2014

Dedicated to my parents.

Love you forever. Thank you for your lift-time support. I want to dedicate this thesis to my lovely parents (Jibiao He and Liyun Yu) for believing, supporting and always being there for me.

ABSTRACT

SHAPE OPTIMIZATION OF AIRFOILS WITHOUT AND WITH GROUND EFFECT USING A MULTI-OBJECTIVE GENETIC ALGORITHM

by

Yilei He

Master of Science in Mechanical Engineering

Washington University in St. Louis, 2014

Research Advisor: Professor Ramesh K. Agarwal

The focus of this thesis is on shape optimization using a genetic algorithm. A multi-objective genetic algorithm (MOGA) is employed to optimize the shape of an airfoil to improve its lift and drag characteristics, in particular to achieve two objectives simultaneously that is to increase its lift as well as its lift to drag ratio. The commercially available software FLUENT is employed to calculate the flow field on an adaptive structured mesh, which is generated by the commercial mesh generating software ICEM. The flow field is calculated using the Reynolds-Averaged Navier-Stokes (RANS) equations in conjunction with a two equation SST $k-\omega$ turbulence model. Bezier Curves are employed to generate airfoil shapes for a particular generation of the genetic algorithm; these shapes are tested by MOGA in conjunction with FLUENT to evaluate their fitness by calculating their lift and lift to drag ratio. The process is continued for a number of generations until the lift and lift to drag ratios converge to their optimal values. MOGA optimization method is used to optimize a well-known wind turbine airfoil S809 and NACA 4412 airfoil in ground effect. The results show significant improvement in both the lift coefficient and lift-to-drag ratio of the optimized airfoil compared to the original airfoil.

Chapter 1

Introduction

1.1 Motivation

Aerodynamic design optimization has been a subject of extensive investigation in aviation industry since it can effectively bring improvements to various desirable aspects of aircraft design, such as lighter weight, better aerodynamic properties and improved fuel-efficiency.

In recent years, aerodynamic shape optimization has also been increasingly used in the design of wind turbines. With current emphasis on emission free renewable energy, wind energy has taken a center stage in recent years with exponential growth in deployment of wind turbines worldwide. Among wind-turbines, Horizontal-Axis-Wind-Turbines (HAWTs) are the most deployed for power generation in Megawatt range. It is well established that the power generated by a HAWT is a function of the number of blades; the Cl/Cd of the blade airfoil section and the tip speed ratio λ (= rotational speed of the blade at tip/wind speed in free stream). Thus, one of the goals of the efficient aerodynamic design of a wind turbine blade is to maximize its Cl/Cd .

Wing in ground effect (WIG) aircraft is a type of aircraft that operates under a very limited ground clearance. It is more efficient for short take-off condition than a standard transport aircraft and is much quicker in response than other equivalent marine vessels [1]. These two advantages offer the WIG craft a promising future in both military and civilian applications. Thus, it is desirable to perform airfoil design optimization for wing in ground effect aircraft in close vicinity of the ground to improve its aerodynamic properties. The aerodynamic optimization criteria for WIG craft are also increasing in lift and lift to drag ratio in ground effect conditions.

1.2 Brief Review of Literature

In modern wind turbines, traditionally thick airfoils such as NACA-63XXX and NACA-64XXX series of airfoils have been frequently employed, however several new airfoil families are being developed because of multiple design requirements that include both the aerodynamic and structural optimization. The aerodynamic optimization requires consideration of aerodynamics performance of wind turbine at both the rated power conditions and the off-rated power conditions. The structural optimization requires consideration of strong structural properties under high and stormy wind conditions [2]. National Renewable Energy Laboratory (NREL) in Colorado has developed a family of airfoils for HAWT applications [3] since 1984. Among them, S809 is a widely-used airfoil. This airfoil is a 21% thick laminar flow airfoil whose profile data and aerodynamic properties for several wind conditions (determined experimentally) are given in Reference [4]. NREL Phase II, Phase III and Phase VI HAWT consist of S809 airfoil from root to tip [5]. The shape optimization of NREL S809 has been recently reported by Ritlop and Nadarajah [6], they employed the adjoint equation based optimization method in study. The focus of this thesis is on using the multi-objective genetic algorithm (MOGA) for shape optimization of NREL S809 airfoil and compare the results with those obtained by Ritlop and Nadarajah [6]. The idea is to compare the two distinct approaches for shape optimization, it has never been done before.

The flow around an aircraft is forced to be parallel to the ground due to ground effect when the aircraft is flying in close proximity to the ground. Thus, the aerodynamics of the aircraft in ground effect is significantly different from that is out of ground effect [7, 8]. In the realm of ground effect aircraft aerodynamics, most studies were focused on steady ground effect (SGE), wherein the flying altitude does not vary with time while a few have investigated the dynamic ground effect (DGE), wherein the flying altitude varies continuously with time [9-16]. There is hardly any study reported in the literature to date on shape optimization of a WIG craft airfoil in ground effect. This is a focus of this thesis. The goal is to optimize a widely-used airfoil for ground effect aircraft, namely NACA 4412 airfoil using a multi-objective genetic algorithm (MOGA) to improve its aerodynamic characteristics. In addition, the optimization study of NACA 4412 airfoil in free stream is also conducted. The airfoils optimized under free stream condition and the ground effect condition (with ground clearance of

0.2C) are compared to explore whether it is necessary to shape optimize an airfoil for a ground effect aircraft to improve its overall aerodynamic performance without and with ground effect.

1.3 Scope of the Thesis

The thesis addresses two problems in aerodynamic shape optimization with two different applications. The first problem addresses the shape optimization of wind turbine airfoil S809 and the second problem addresses the shape optimization of NACA 4412 in ground effect. A multi-objective genetic algorithm is employed for optimization. The flow field is computed using the commercial CFD software FLUENT which solves the Reynolds-Averaged Navier-Stokes (RANS) equations in conjunction with a two equation SST k- ω turbulence model. Airfoil shapes needed in a given generation of genetic algorithm are generated by the Bezier Curve. Grid around the airfoils is automatically generated by the mesh generator software ICEM. Both FLUENT and ICEM are part of ANSYS software.

Chapter 2

Optimization Methodology and Flow Field Simulations of Original S809 and NACA 4412 Airfoils

This section presents the flow field simulation and optimization methodology for S809 wind turbine airfoil and the NACA 4412 airfoil in ground effect. An optimization procedure is established by coupling the Multi-Objective Genetic Algorithm (MOGA) based optimization code with the mesh generation software ANSYS-ICEM and the CFD solver ANSYS-FLUNET as shown in Fig. 1.

The individuals in each generation of genetic algorithm (GA) are represented by a set of control points, which generate the airfoil shape through the Bezier Curve. The mesh around the airfoil shape is generated using the grid generation software ICEM, which is used to create a two-dimensional structured or unstructured mesh as an input to the CFD solver FLUENT. FLUENT is used to calculate the flow field for given flow conditions. Using the flow field data, FLUENT calculates the lift coefficient C_l and the drag coefficient C_d which are used as the two objective's fitness values for a given airfoil shape. Using the information about the objective values for all the airfoils in a given generation, MOGA is applied to create a next generation of airfoils and the process is repeated to obtain the Pareto front following the MOGA procedure outlined in section 2.1.2. From the Pareto front, optimal solution for objective values is obtained. The airfoil shape that corresponds to the optimal objective values is the final shape of the optimized airfoil [17].

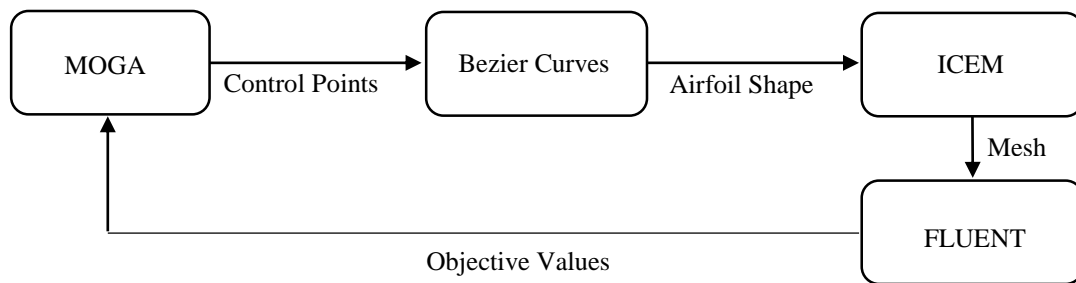


Figure 1 Schematic of information flow in the optimization process

2.1 Genetic Algorithm based Optimization Method

2.1.1 Overview of genetic algorithm

Genetic algorithms (GA) are a class of stochastic optimization algorithms inspired by the biological evolution. In GA, a set of generation of input vectors, called individuals, is iterated over, successively combining traits (aspects) of the best individuals until a convergence is achieved. In general, GA employs the following steps [18, 19].

1. **Initialization:** Randomly create N individuals.
2. **Evaluation:** Evaluate the fitness of each individual.
3. **Natural selection:** Remove a subset of the individuals. Often the individuals that have the lowest fitness are removed; although culling, the removing of those individuals with similar fitness, is sometimes performed.
4. **Reproduction:** Pick pairs of individuals to produce an offspring. This is often done by roulette wheel sampling; that is the probability of selecting some individual h_i for reproduction is given by:

$$P[h_i] = \frac{fitness(h_i)}{\sum_j fitness(h_j)} \quad (1)$$

A crossover function is then performed to produce the offspring. Generally, crossover is implemented by choosing a crossover point on each individual and swapping alleles – or vector elements – at this point as illustrated in Fig. 2.

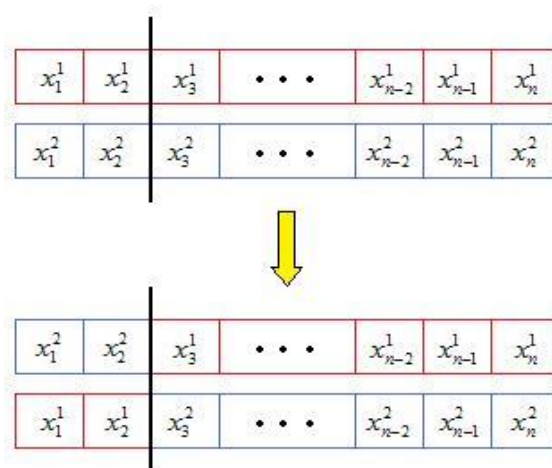


Figure 2 Illustration of the general Crossover function in GA

5. **Mutation:** Randomly alter some small percentage of the population.
6. **Check for Convergence:** If the solution has converged, return the best individual observed. If the solution has not yet converged, label the new generation as the current generation and go back to step 2. Convergence often occurs after a certain number of generations meeting a user defined tolerance criteria for acceptable change in objective values from one generation to next.

2.1.2 Multi-objective Genetic Algorithm (MOGA)

For many design problems, it is desirable to achieve simultaneous optimization of multiple objectives [20]. Sometimes, these objectives, however, maybe conflicting, preventing simultaneous optimization of each objective [21]. Therefore, instead of searching for a single optimal solution, a multi-objective genetic algorithm (MOGA) is applied to find a set of optimal solutions (generally known as Pareto-optimal solutions). In Pareto-optimal solutions, individual inside the set may dominate individual outside the set while individual in the set is not dominated by another individual in the solution set. The MOGA algorithms employed to find the Pareto-optimal solutions for the airfoil optimization problems considered in this study is widely known as the NSGA-II. It has the following three features: (1) it uses an elitist principle, (2) it uses an explicit diversity preserving mechanism, and (3) it emphasizes non-dominated solutions in a population [22]. The implementation procedure of NSGA-II is as follows [17]:

1. At **0-th generation**, a random parent population P_0 of size N is created; it is sorted based on the non-domination. Then the individuals in P_0 are ranked: 1 is the best level, 2 is the next-best level, and so on. Then P_0 is sent to selection, recombination, and mutation operators to create off-spring population Q_0 of size N .
2. At **t-th generation**, a combined population $R_t = P_t \cup Q_t$ of size $2N$ is formed and is sorted according to non-domination. Then individuals in R_t are divided into the best non-dominated set F_1 , the next-best non-dominated set F_2 and so on. If the size of F_1 is smaller than N , all members of F_1 go to P_{t+1} , with the remaining members chosen from $F_2, F_3 \dots$ until the size of P_{t+1} is N . Then new population P_{t+1} is sent to selection, crossover, and mutation operators to create a new population Q_{t+1} of size N .
3. **Termination**, the procedure terminates when convergence criterion is met.

The java code package utilized in this study is called jMetal. It is a Java-based framework for multi-objective optimization using meta-heuristics. It is easy-to-use and is flexible and extensible [23].

2.1.3 Implementation of multi-objective genetic algorithm (MOGA)

NSGA-II [17] and the jMetal [23] multi-objective GA software packages are employed. We choose 20 individuals (airfoils) for each generation. The crossover rate of 0.9 is considered. The mutation rate is determined to be 1/24. jMetal MOGA framework offers multiple operators, here we employ the simulated binary crossover (SBX) operator and the polynomial mutation operator for crossovers and mutations respectively. The selection process employs the binary tournament operator.

The multi-objective optimization algorithm is performed with two objective functions. The first objective is to minimize $10/Cl$, and the second objective is to minimize $100 \cdot Cd/Cl$. The goal is to find the Pareto front for these two objective functions. When the value of both objective functions does not change from one generation to the next, the solution is considered converged to an optimal value of Cl and Cl/Cd . The airfoil shape that corresponds to the optimal objective values is the final shape of the optimized airfoil.

2.2 Airfoil Parameterization

The airfoil shapes are parameterized using Bezier curves. Bezier curves are parametric curves frequently used in computer graphics and related fields. A Bezier curve is defined by a set of Bezier control points. Each curve can be expressed as math equations containing the information of Bezier control points. The number of control points required to parameterize a curve depends on the shape of the curve.

Each airfoil is divided into top and bottom boundary curves by the airfoil chord joining its leading edge and the trailing edge. Considering the shape complexity of S809 and NACA 4412 airfoil, 12 control points are used for parameterization. For an airfoil curve, two points are fixed since they represent the leading and trailing edge of the airfoil. The intermediate points are allowed to move within the specified constraints. A maximum thickness constraint of 19%-22% of chord is imposed on S809 and 16%-17% on NACA 4412. The constraints applied to the Bezier control points are shown in Table 1 and Table 2. Figure 3 shows the distribution of control points on S809 airfoil.

Table 1 Coordinates of the Bezier curve control points for S809 airfoil parameterization

		Upper Limit	Lower Limit			Upper Limit	Lower Limit
Top Boundary	x1	0.020	0.000	Bottom Boundary	m1	0.020	0.000
	x2	0.065	0.025		m2	0.100	0.060
	x3	0.340	0.300		m3	0.300	0.260
	x4	0.400	0.360		m4	0.400	0.360
	x5	0.500	0.460		m5	0.480	0.440
	x6	0.850	0.810		m6	0.770	0.730
	y1	0.030	0.010		n1	-0.013	-0.025
	y2	0.090	0.070		n2	-0.053	-0.065
	y3	0.128	0.108		n3	-0.125	-0.145
	y4	0.128	0.108		n4	-0.125	-0.145
	y5	0.128	0.108		n5	-0.125	-0.145
	y6	0.045	0.025		n6	0.020	0.008

Table 2 Coordinates of the Bezier curve control points for NACA 4412 airfoil parameterization

		Upper Limit	Lower Limit			Upper Limit	Lower Limit
Top Boundary	x1	0.0054	0.0050	Bottom Boundary	m1	0.0012	0.0010
	x2	0.040	0.038		m2	0.042	0.040
	x3	0.296	0.294		m3	0.271	0.269
	x4	0.367	0.365		m4	0.411	0.410
	x5	0.452	0.450		m5	0.394	0.393
	x6	0.817	0.815		m6	0.764	0.762
	y1	0.016	0.014		n1	-0.012	-0.018
	y2	0.095	0.093		n2	-0.020	-0.026
	y3	0.113	0.111		n3	0.005	0.0011
	y4	0.149	0.147		n4	-0.015	-0.03
	y5	0.159	0.157		n5	0.010	0.014
	y6	0.049	0.047		n6	0.0061	0.0066

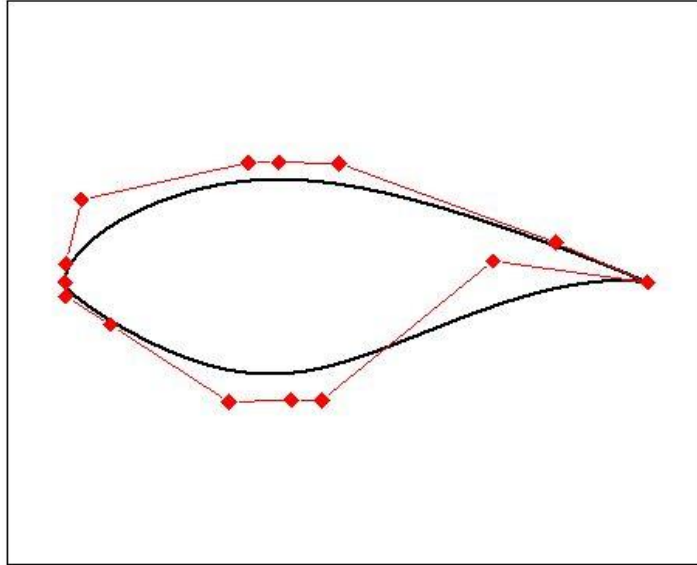


Figure 3 Distribution of Bezier-curve control points on S809 airfoil

2.3 Mesh Generation

The commercially available software “ICEM” is used to generate a structured mesh around the S809 and NACA 4412 airfoils. Adaptive meshing is employed. A reply file is scripted to automatically generate mesh around different airfoils in a given generation. The reply file is edited to be able to generate mesh based on different airfoil shapes. Figure 4 shows a typical C-mesh around the S809 airfoil. In this mesh, there are 58460 quadrilateral cells. Far field boundary is set at 20 chord lengths. Figure 5 shows the structured mesh around the NACA 4412 airfoil. Approximately 50305 quadrilateral cells exist in this computational domain. The ground is treated as a moving wall boundary.

Grid independence of the solution for both the optimization cases is assessed by computing solution on another finer mesh. The number of quadrilateral elements is doubled in the finer mesh. The height of the first layer element in the finer mesh is half of that in the original mesh. The results show that the variation in lift coefficient resulting from the change in number of grid points both in the cases is within the acceptable range of accuracy. Thus, the original mesh with cell numbers mentioned above is adopted.

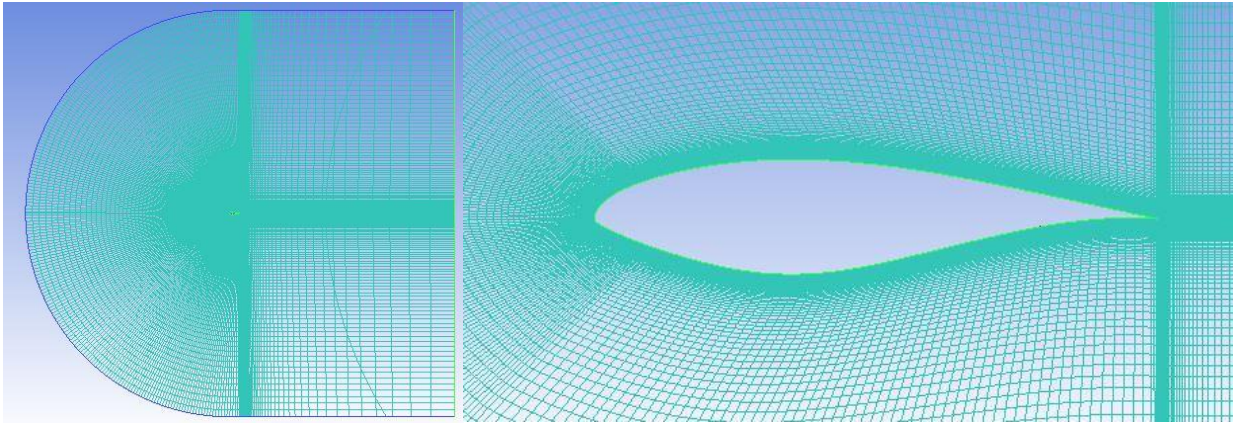


Figure 4 Structured C-grid mesh around the NREL S809 airfoil

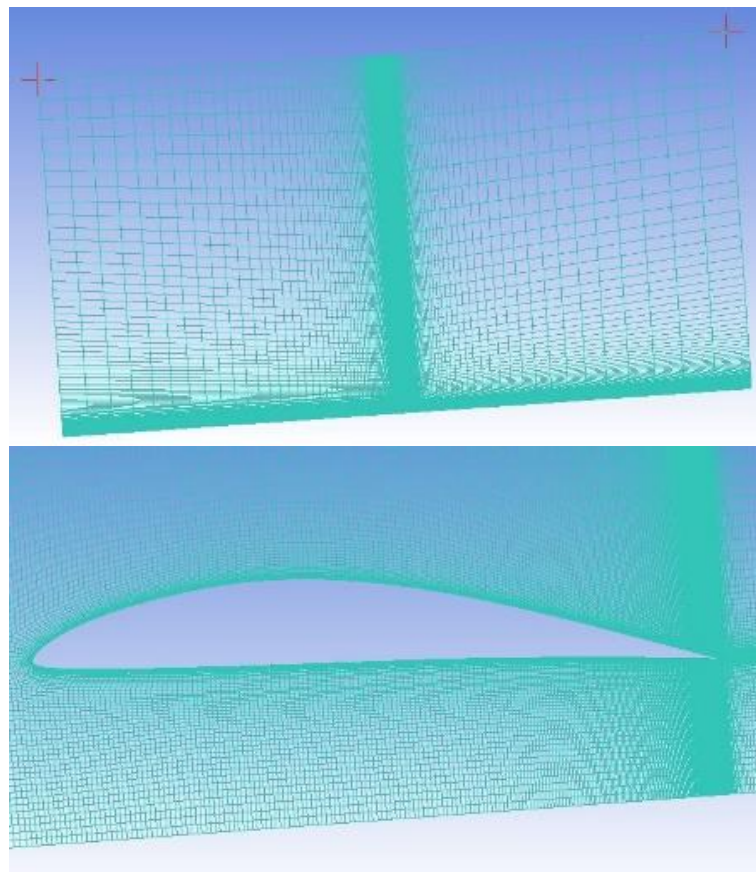


Figure 5 Structured Mesh around the NACA 4412 airfoil in ground effect

2.4 Flow Field Computations on Original S809 Airfoil and Tyrrell-026 Airfoil in Ground Effect for Code Validation

The commercial software FLUENT is employed to calculate the lift coefficient and the drag coefficient of the original S809 airfoil and Tyrrell-026 airfoil in ground effect for code validation and to demonstrate the grid independence of the solutions. A journal file is written for auto-running of the FLUENT in the optimization process described in Chapter 3.

2.4.1 Wind-Turbine S809 Airfoil

The typical Mach number encountered by a wind turbine airfoil is around $M = 0.1$. Because of very low Mach number, the convergence of the flow solver is very slow. It is due to the well-known fact that the time-marching flow solvers designed for compressible flows do not perform well at very low subsonic Mach numbers due to large condition number of the associated eigenvalue matrix [24]. To address the problem of slow convergence, we multiply the free stream velocity of the wind by some numbers which increases the Mach number but still keeps it in the low subsonic flow regime. We keep the Reynolds number of the flow unchanged by dividing the chord length of the airfoil by the numbers. We validate our calculations by comparing them with the numerical results of Ritlop and Nadarajah [6] and the experimental data of Ramsey [25]. The simulations are performed at Reynolds number of 1 million and Mach number of 0.044. Nine different angles of incidence are selected for validation, which are 0.0° , 2.1° , 4.1° , 6.1° , 8.2° , 10.1° and 11.2° . The turbulence model employed is a two-equation k-omega SST turbulence model. For the calculations, temperature and static pressure are defined at standard sea level condition and are taken as 288.16 K and 101325 Pa respectively. Both values are quite reasonable for a wind turbine whose maximum altitude does not exceed a few hundred meters. Density is taken as $\rho = 1.225 \text{ kg/m}^3$ and the laminar viscosity is taken as $\mu = 1.7894 * 10^{-5} \text{ kg/m.s}$. Tables 3 and 4 and Figs. 6 and 7 show that our computed results for lift and drag coefficient agree reasonably well with the experimental values [25] as well as with the computational results of Ritlop and Nadarajah [6]. These comparisons validate the numerical methodology used in our calculations for computing flows at very low Mach number.

Table 3 Computed lift coefficients for S809 airfoil and their comparison with experimental lift coefficients

AoA [deg]	Comp. Cl	Exp. Cl
0.0	0.12529	0.07
2.1	0.35428	0.3
4.1	0.55472	0.55
6.1	0.75412	0.79
8.2	0.94169	0.9
10.1	1.0678	0.94
11.2	1.1046	0.93

Table 4 Computed drag coefficients for S809 airfoil and their comparison with experimental drag coefficients

AoA [deg]	Comp. Cd	Exp. Cd
0.0	0.012006	0.0022
2.1	0.012824	0.0037
4.1	0.015248	0.005
6.1	0.017615	0.0063
8.2	0.021507	0.0096
10.1	0.027757	0.0231
11.2	0.03413	0.0236

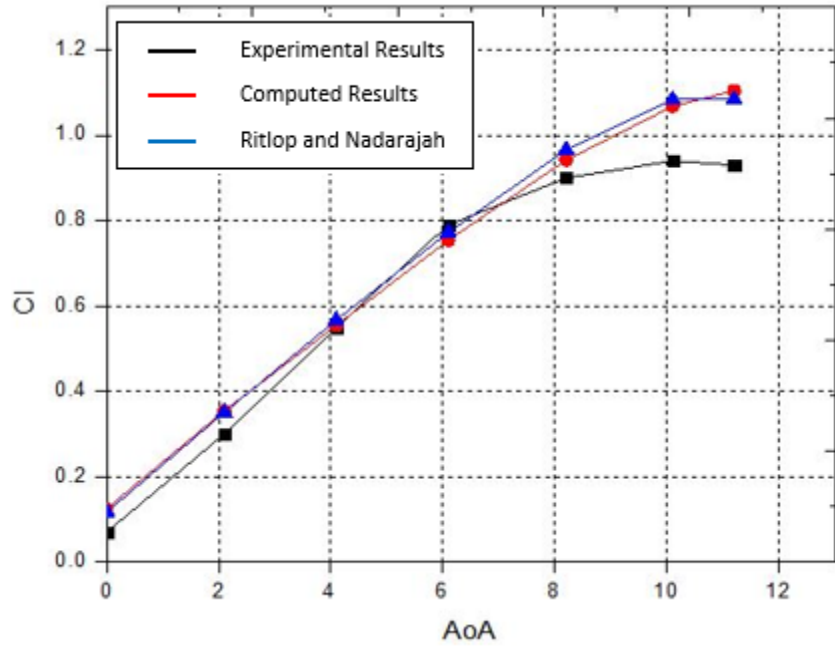


Figure 6 Variation in lift coefficient with angle of attack

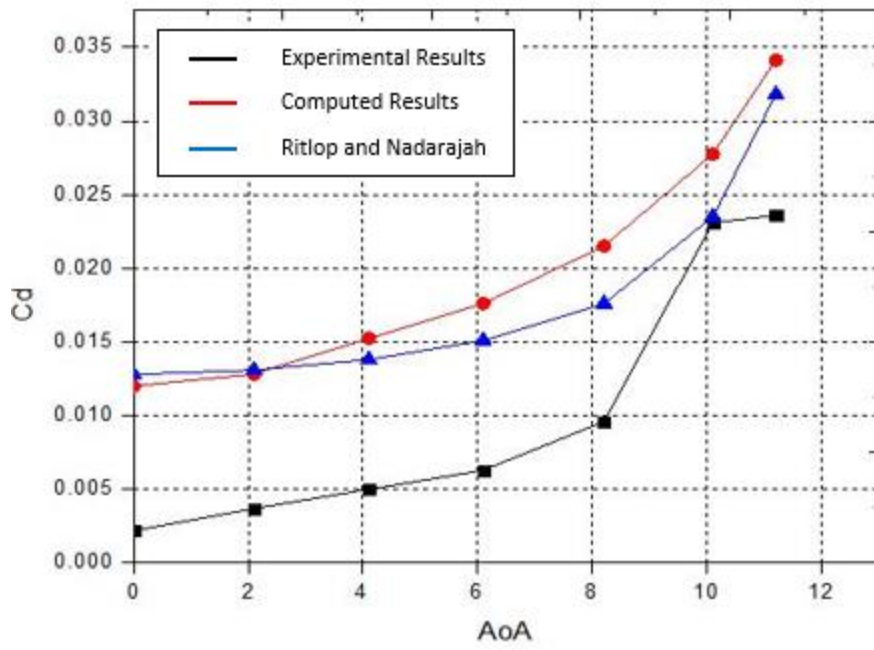


Figure 7 Variation in drag coefficient with angle of attack

2.4.2 Tyrrell-026 Airfoil in Ground Effect

In order to evaluate the accuracy of the ANSYS-FLUENT code for predicting the aerodynamic forces on an airfoil in ground effect, the flow field of a Tyrrell-026 airfoil in static ground effect (SGE) is calculated and compared with the wind tunnel experiment data [11]. Figure 8 shows the pressure coefficient distribution from our CFD simulation which matches the experimental data quite well.

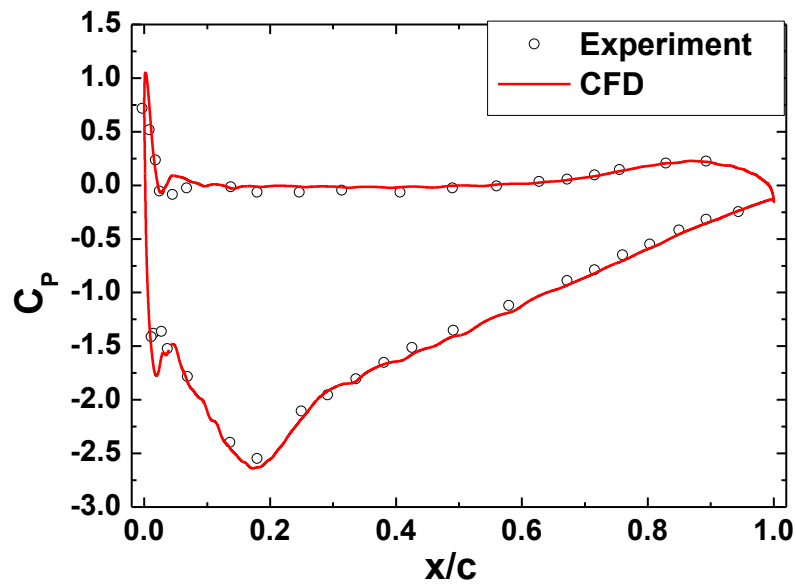


Figure 8 Comparison of computed and experimental pressure coefficient for the Tyrrell-026 airfoil

Chapter 3

Optimization Results

3.1 Results for optimized S809 wind turbine airfoil

Wind turbines generate power due to rotation of the blades in streaming wind. Blade Element Momentum (BEM) theory is used to determine the generated power [26]. The BEM theory is based on the Glauert's propeller theory [27] which has been modified for application to wind turbines. In this thesis we are interested in determining the relative velocity faced by an airfoil of the wind turbine as shown in Fig. 9 from Reference [28]. The expressions for axial velocity and blade rotational velocity at blade radius r are given by the axial induction factor a and the radial or rotational induction factor a' respectively. The relative velocity in Fig. 9 can be expressed as [28]:

$$V_{rel} \sin \phi = V_{axial} = V_0(1 - a) \quad (2)$$

$$V_{rel} \cos \phi = V_{blade} = \omega r(1 + a') \quad (3)$$

where ω is the angular velocity of the wind turbine blade, r is the radial position of airfoil section and V_0 is the free stream velocity.

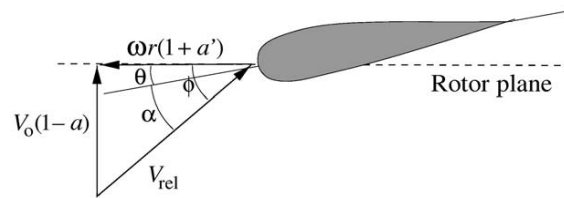


Figure 9 Velocities in the rotor plan [28]

Using the published data in Reference [5], the effective angles of attack α_{eff} and the relative velocity $V_{relative}$ for free stream wind velocity $V_\infty = 6.7m/s$, rotational speed of the turbine $\omega = 72$ rpm and the pitch setting of 5 deg. can be obtained as given in Table 5.

Table 5 Effective angle of attack and relative velocity experienced by the wind turbine airfoil for free stream wind velocity $V_\infty = 6.7\text{m/s}$, rotational speed $\omega = 72\text{ rpm}$ and pitch setting of 5 deg.

Angle of Attack	α_{eff}	$v_{relative}$
0.23	4.871484	10.95754347
0.28	5.33559	12.49505605
0.33	5.507533	14.12325328
0.38	5.695772	15.81257285
0.43	5.65495	17.54835289
0.48	5.481622	19.31840765
0.53	5.285769	21.11286633
0.58	5.113385	22.9245775
0.63	4.943813	24.74883296
0.68	4.799144	26.58353782
0.73	4.692116	28.4262618
0.78	4.619902	30.27473603
0.83	4.540881	32.12794747
0.88	4.399174	33.98431527
0.93	4.133288	35.84205712
0.98	3.397045	37.68558576

In the optimization study, we consider the free stream wind velocity of 6.7m/s, rotational speed of 72 rpm and pitch setting of 5 deg. We optimize the S809 airfoil sector at three locations of the blade - 23%, 58% and 98% locations from the center of the rotor which correspond to the root, mid and tip section of the wind blade respectively.

For these three blade locations, the effective angle of attack and relative blade velocity are determined from Table 5. For 23% span location, $\varphi_{eff} = 4.87^\circ$ and $V_{rel} = 10.96\text{m/s}$, for 58% span location, $\varphi_{eff} = 5.11^\circ$ and $V_{rel} = 22.92\text{m/s}$ and for 98% span location $\varphi_{eff} = 3.39^\circ$ and $V_{rel} = 37.68\text{m/s}$. We set two objectives for optimization: minimize $10/Cl$ and $100 \cdot Cd/Cl$. The airfoil shape that results in lowest values of both the objectives is the shape of the optimized airfoil. Figs. 10, 11 and 12

show the evolution process of the airfoil shape at various generations of MOGA at 23%, 58% and 98% span location respectively. Figs. 13, 14 and 15 show the Pareto front for the airfoils at the three span locations of the blade. Figs. 16, 17 and 18 show the comparison of airfoil shapes and pressure coefficients between the optimal airfoils and original airfoils at three different span locations.

A comparison between the present optimized airfoil shapes using MOGA and those obtained by Ritlop and Nadarajah [6] using the adjoint method is shown in Figs. 19, 20 and 21 at three blade sectors. Figs. 22, 23 and 24 show the pressure and velocity contours around the optimized and original S809 airfoil at 23%, 58% and 98% span location respectively. Table 6 and 7 show the comparison of C_l and C_l/C_d for the present optimized airfoils with those obtained by Ritlop and Nadarajah [6]. For the lift coefficient, the present results compare quite well with those in Reference [6]. For the ratio C_l/C_d , the present results give a slightly higher value than those given in Reference [6]. These results show that MOGA can give optimized airfoils that meet the two objectives of minimizing $10/C_l$ and $100C_d/C_l$ quite well; the results are in close agreement with those obtained by the adjoint method.

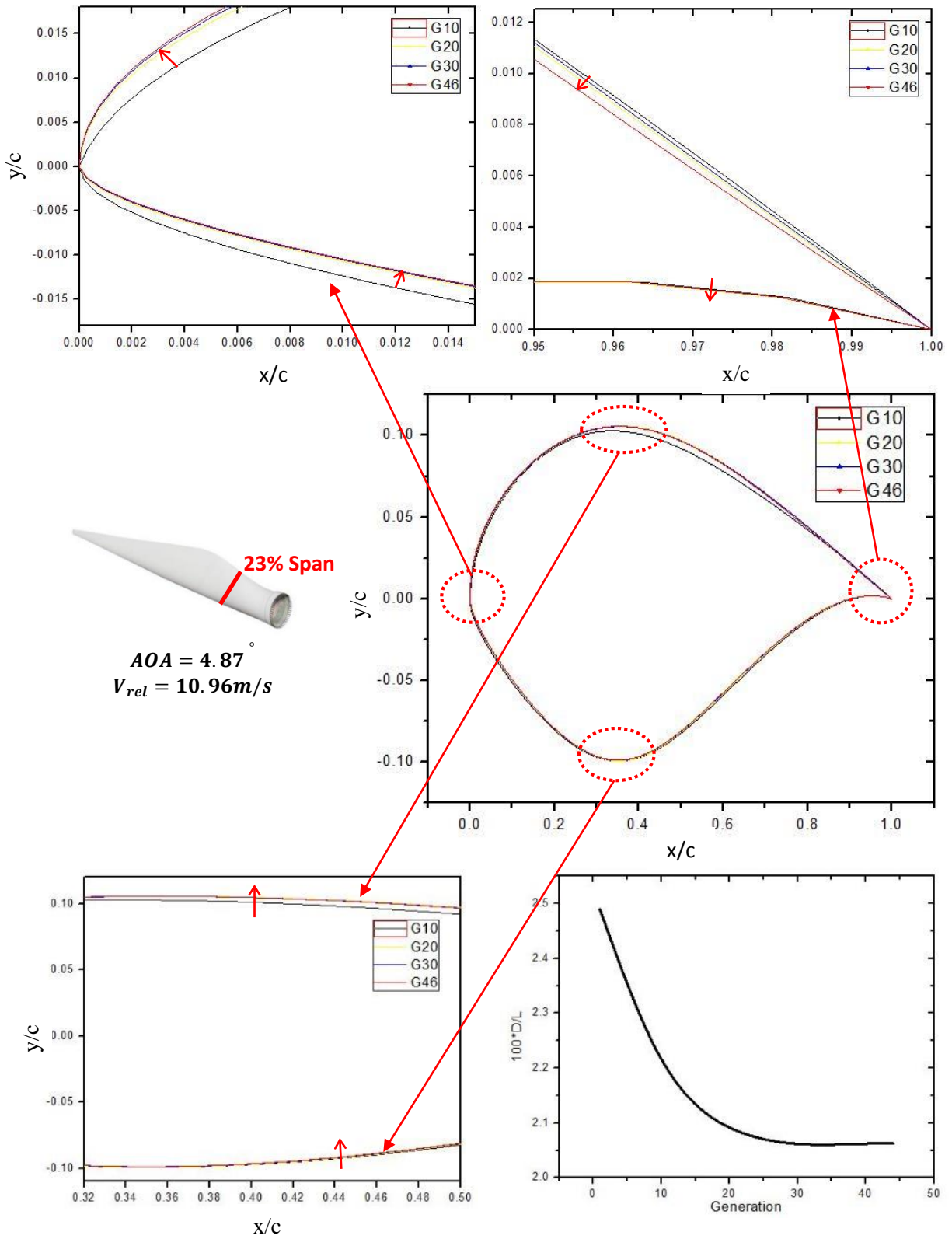


Figure 10 Shape evolution of S809 airfoil at 23% span location at various generations of MOGA

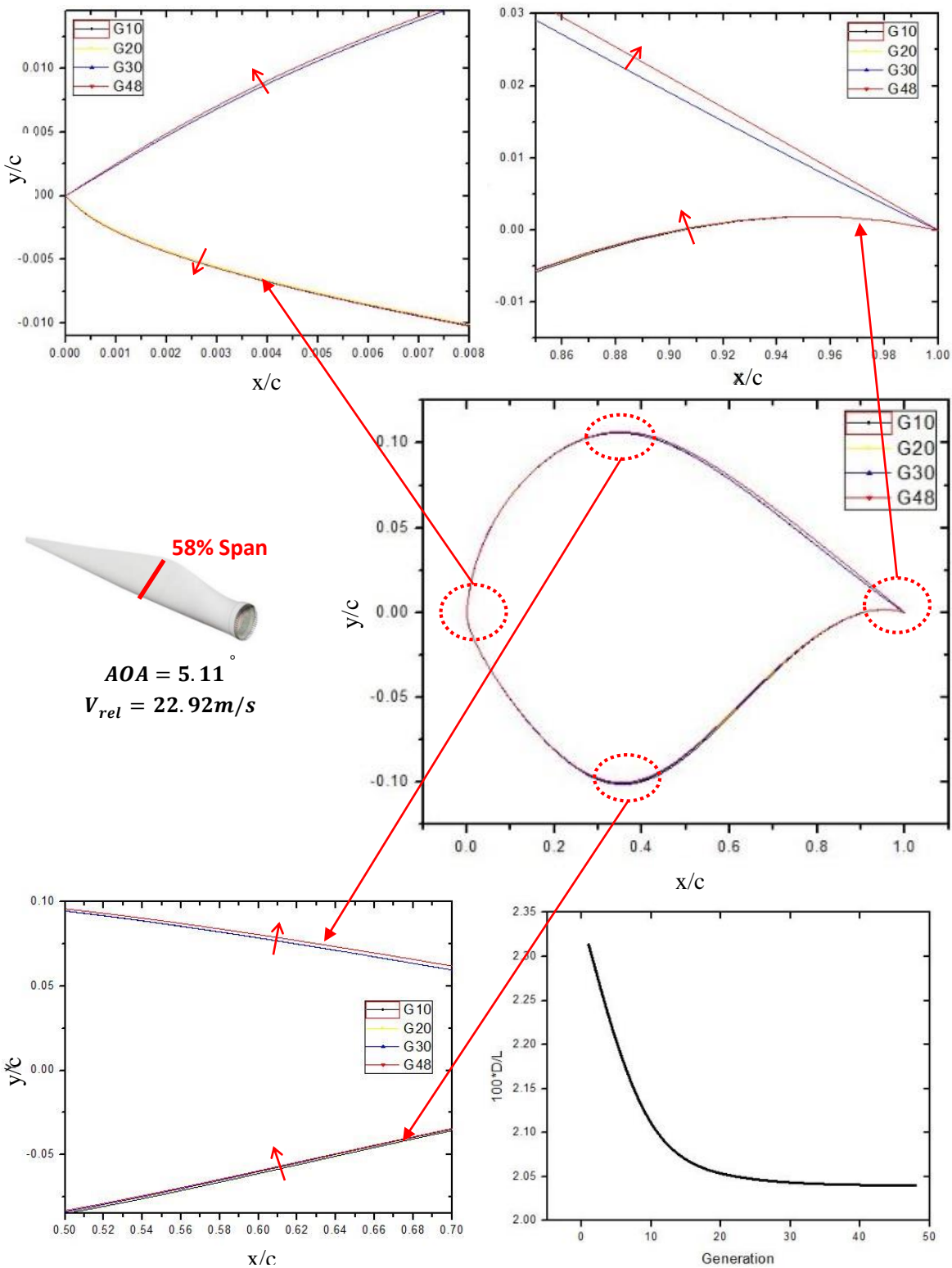


Figure 11 Shape evolution of S809 airfoil at 58% span location at various generations of MOGA

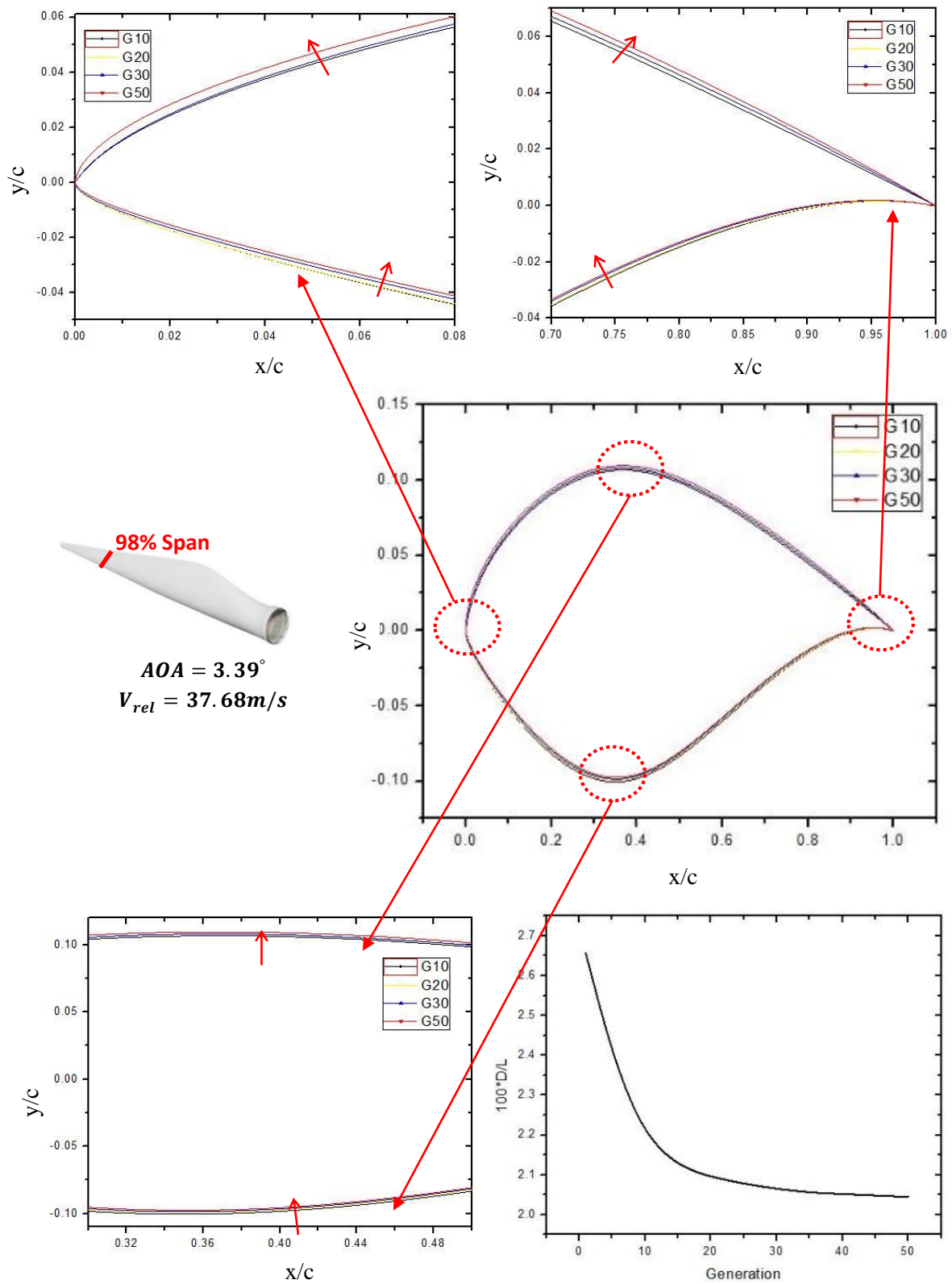


Figure 12 Shape evolution of S809 airfoil at 98% span location at various generations of MOGA

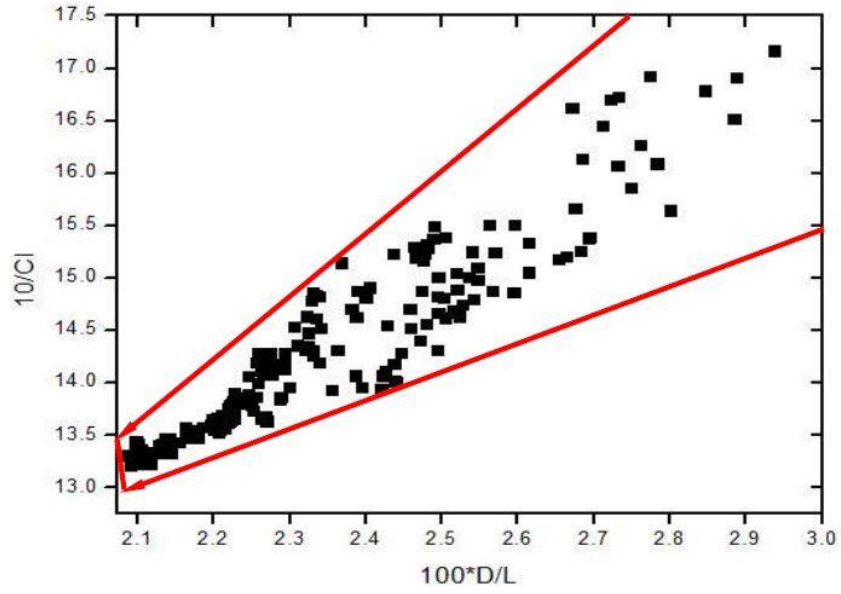


Figure 13 Pareto front for S809 airfoil at 23% span location

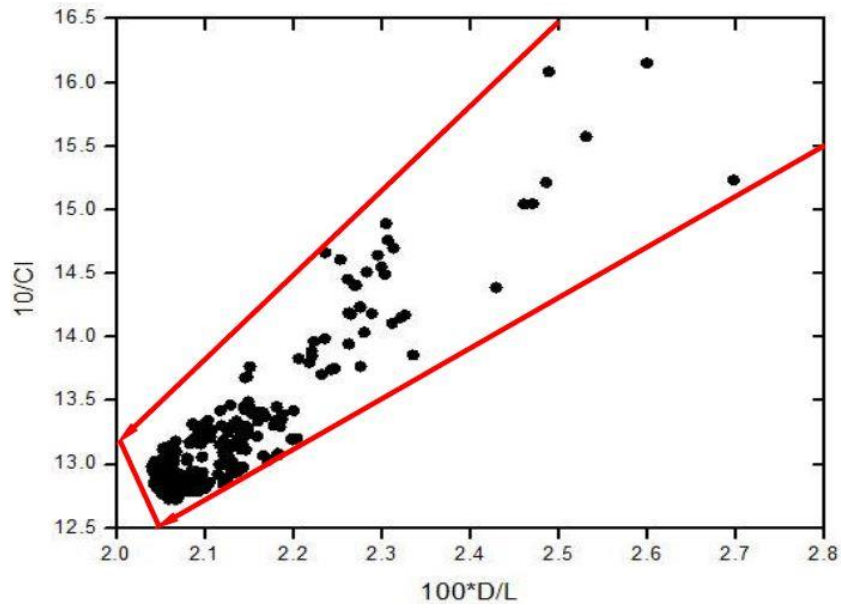


Figure 14 Pareto front for S809 airfoil at 58% span location

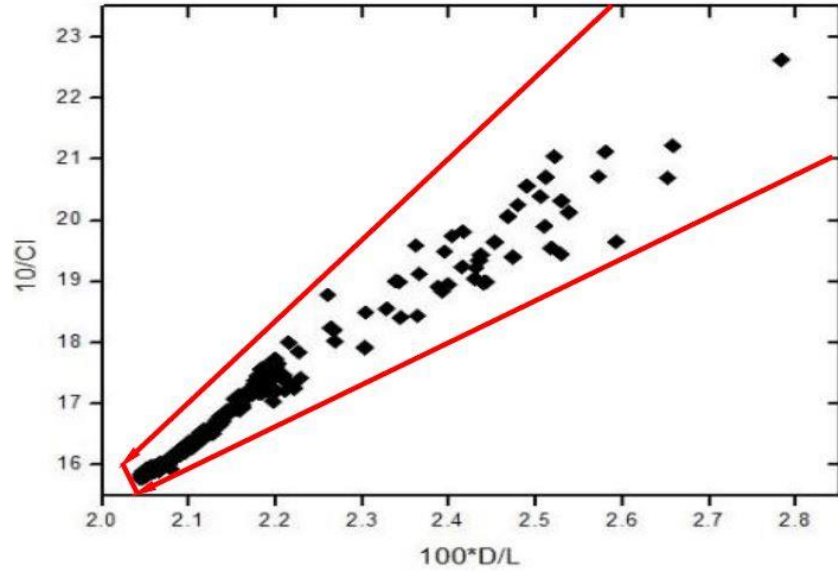


Figure 15 Pareto front for S809 airfoil at 98% span location

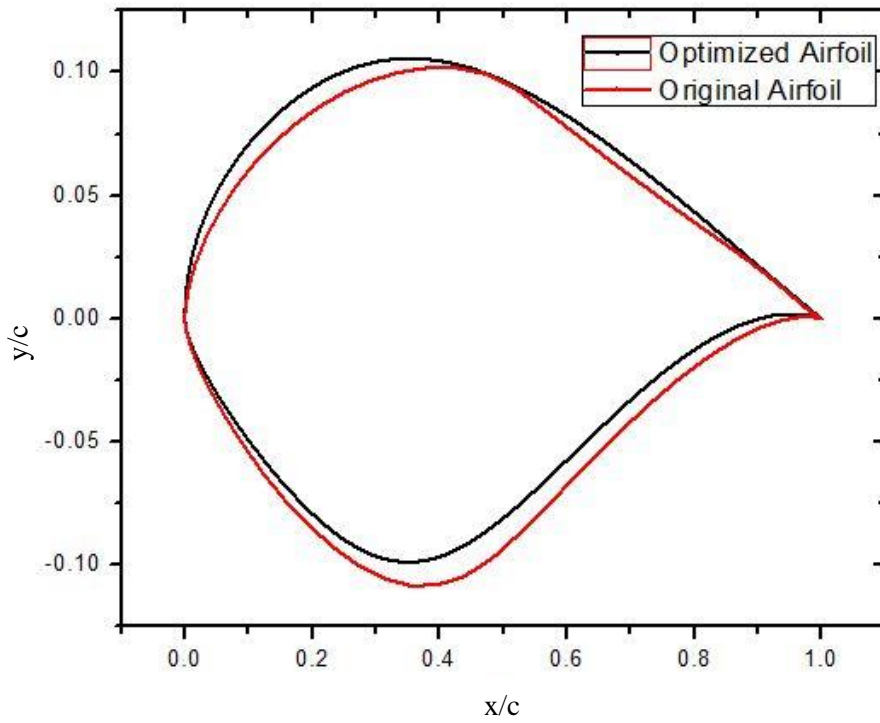


Figure 16(a) Comparison of shape of optimized S809 airfoil and original S809 airfoil at 23% span location

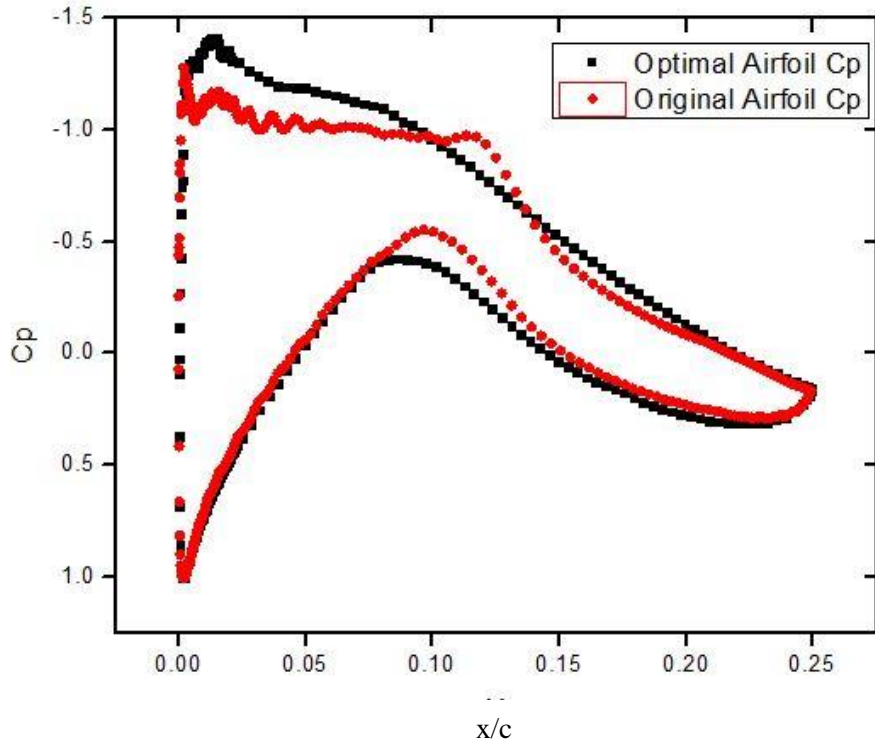


Figure 16(b) Comparison of pressure coefficient of optimized S809 airfoil and the original S809 airfoil at 23% span location

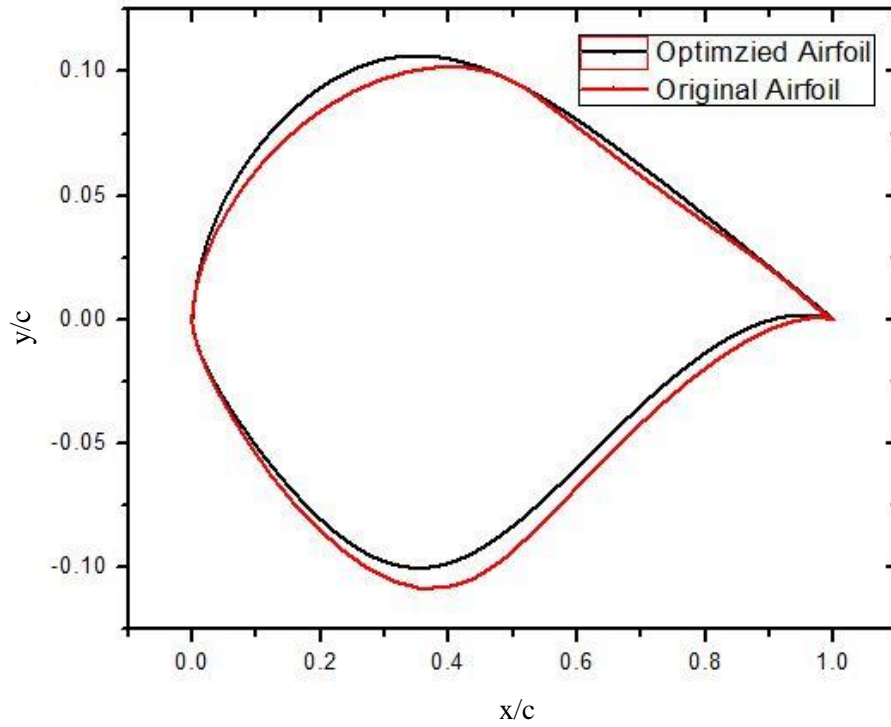


Figure 17(a) Comparison of shape of optimized S809 airfoil and the original S809 airfoil at 58% span location

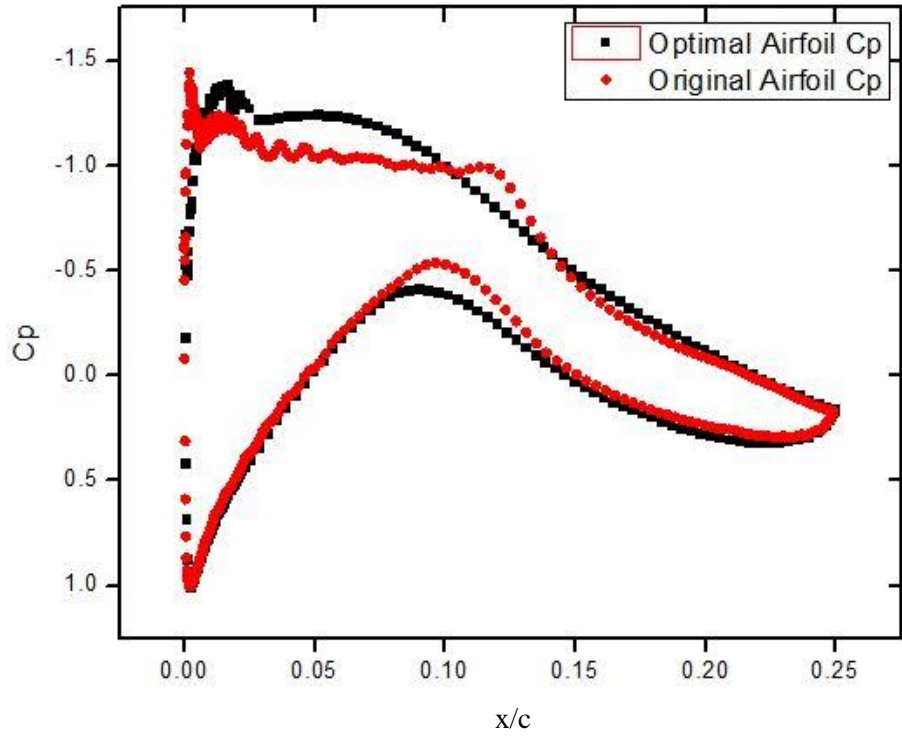


Figure 17(b) Comparison of pressure coefficient of optimized S809 airfoil and the original S809 airfoil at 58% span location

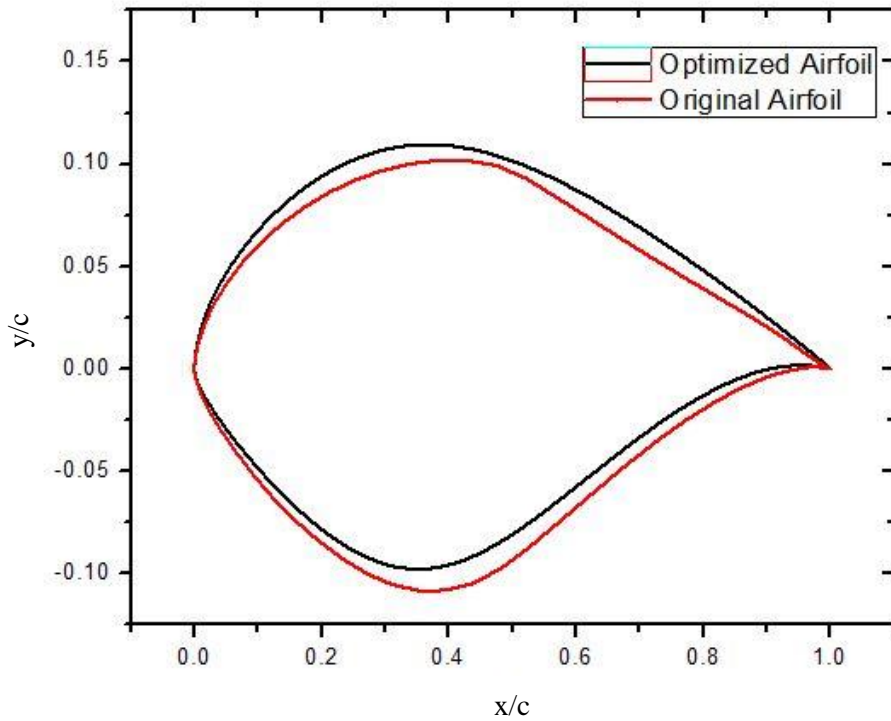


Figure 18(a) Comparison of shape of optimized S809 airfoil and original S809 airfoil at 98% span location

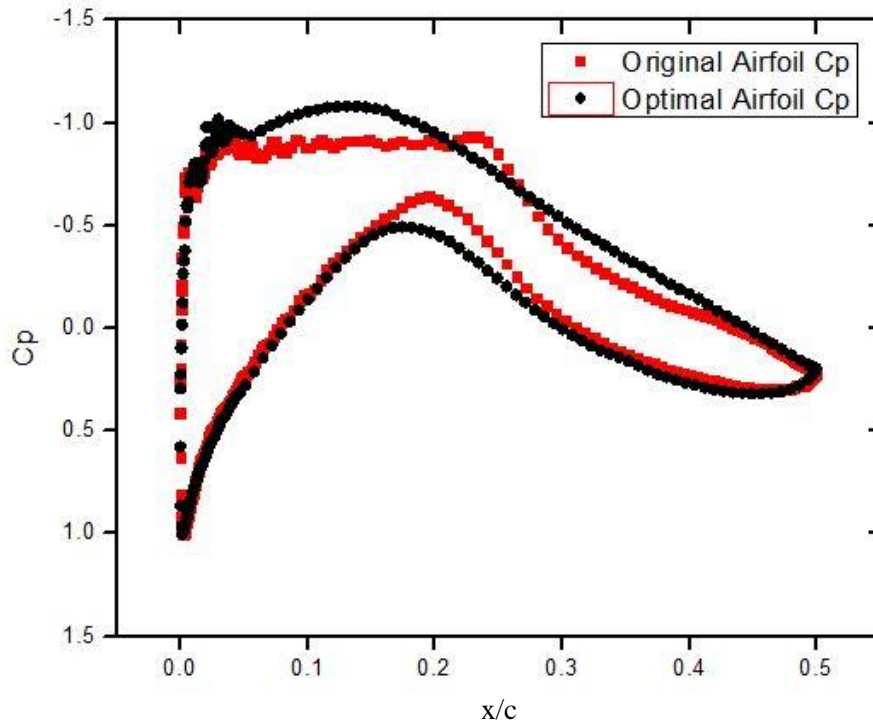


Figure 18(b) Comparison of pressure coefficient of optimized S809 airfoil and the original S809 airfoil at 98% span location

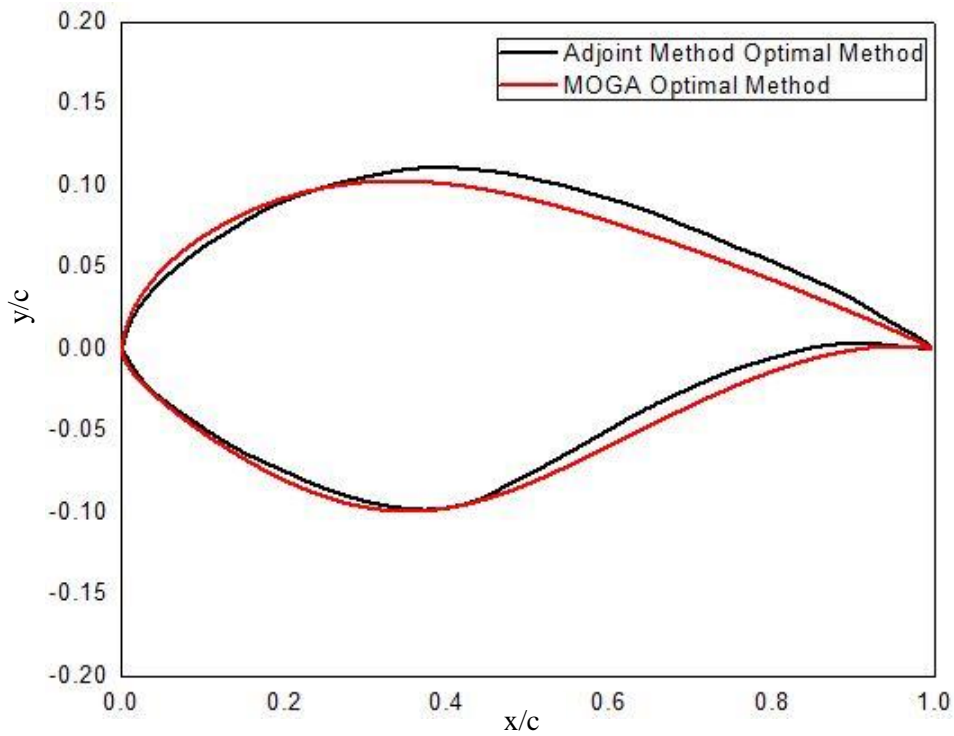


Figure 19 Comparison of MOGA optimized S809 airfoil shape at 23% span location with that obtained by the adjoint method [6]

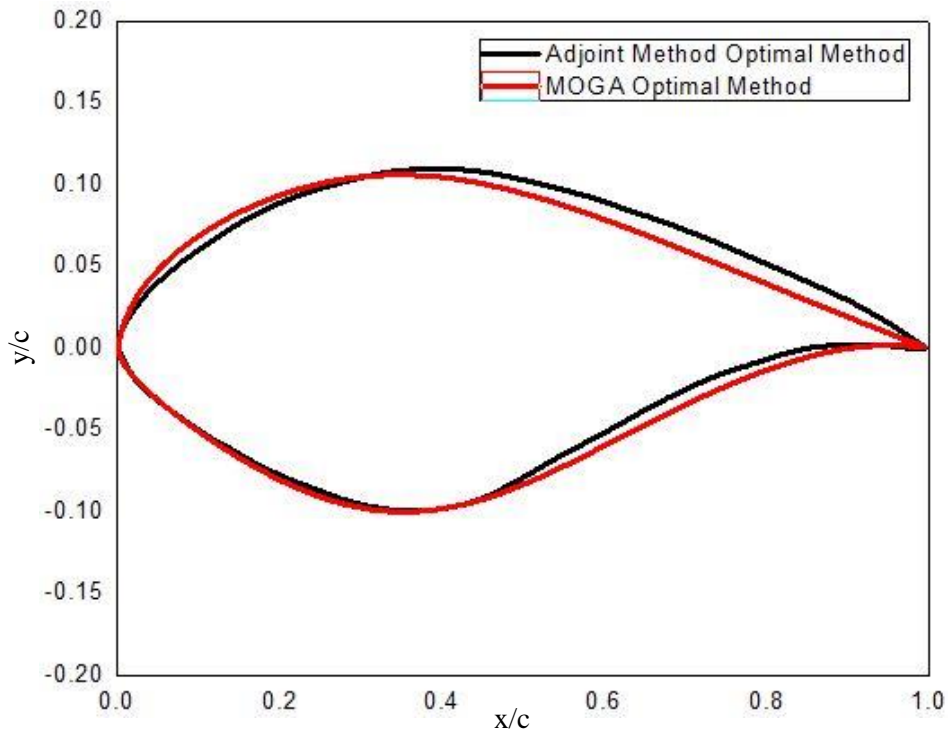


Figure 20 Comparison of MOGA optimized S809 airfoil shape at 58% span location with that obtained by the adjoint method [6]

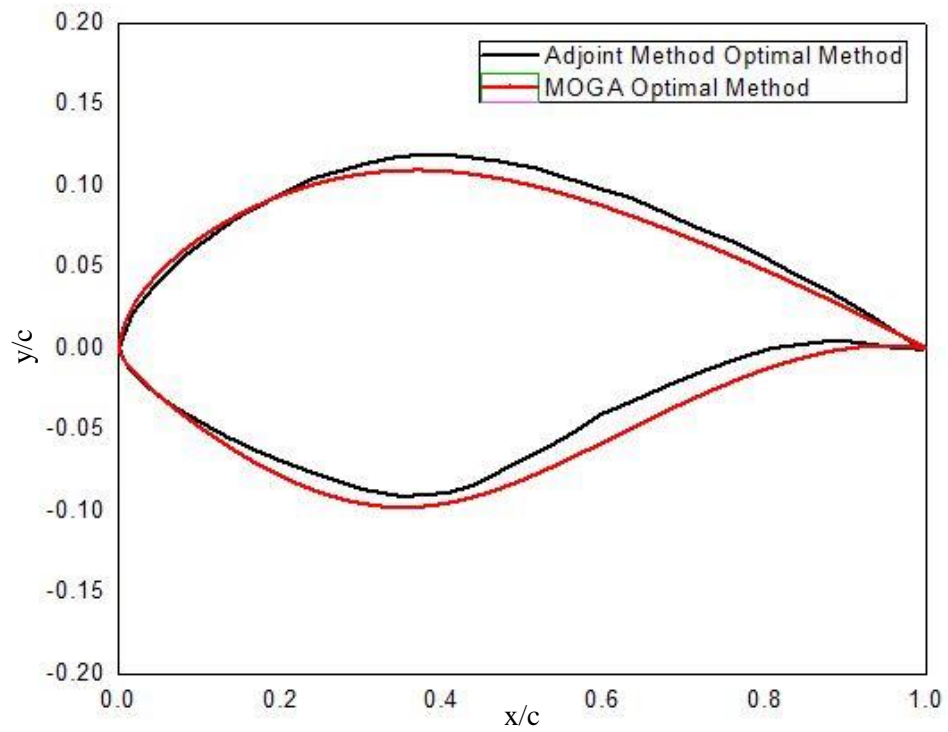


Figure 21 Comparison of MOGA optimized S809 airfoil shape at 98% span location with that obtained by the adjoint method [6]

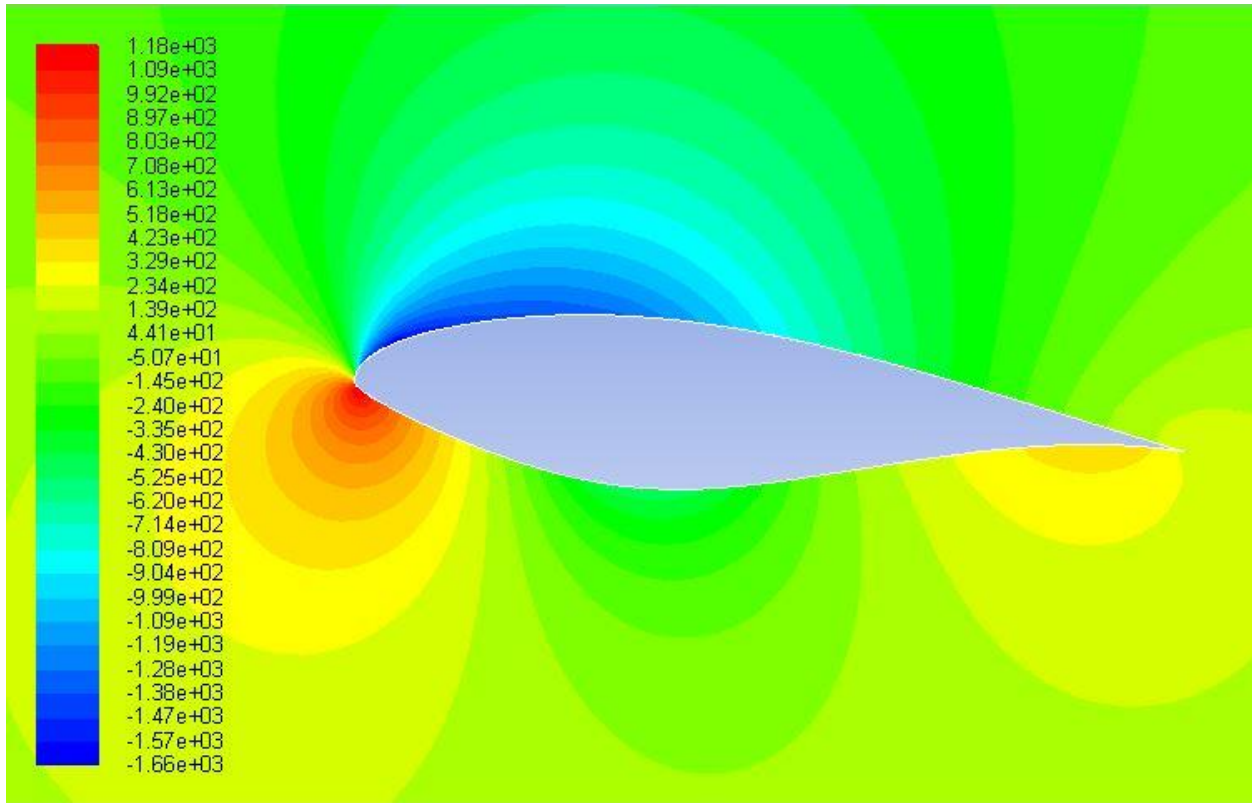


Figure 22(a) Pressure contours around optimized S809 airfoil at 23% span location

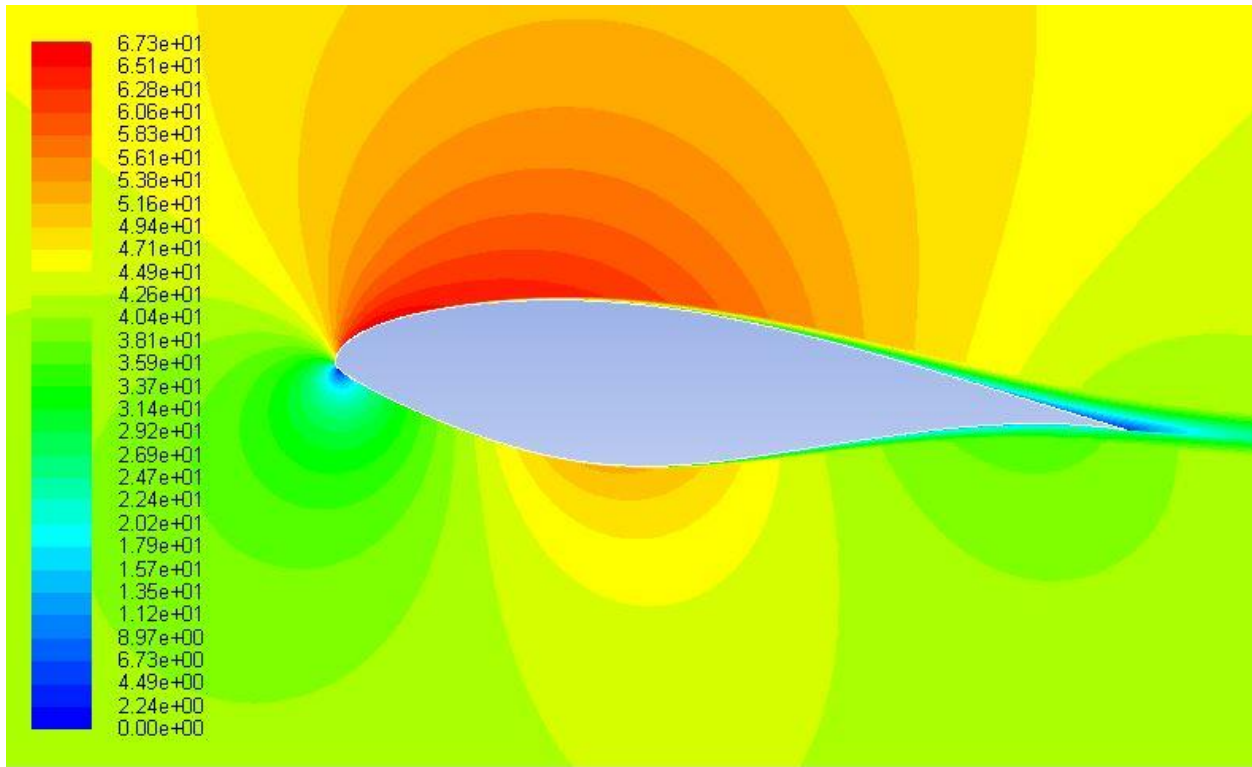


Figure 22(b) Velocity contours around optimized S809 airfoil at 23% span location

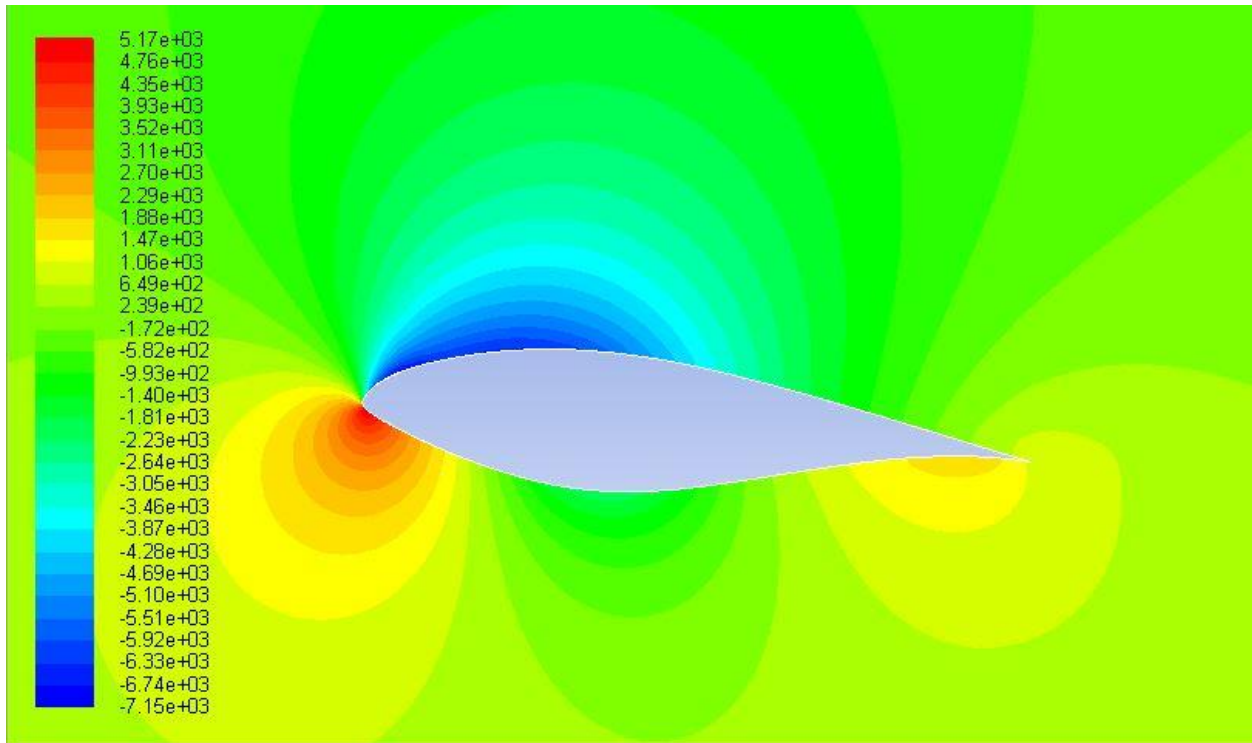


Figure 23(a) Pressure contours around optimized S809 airfoil at 58% span location

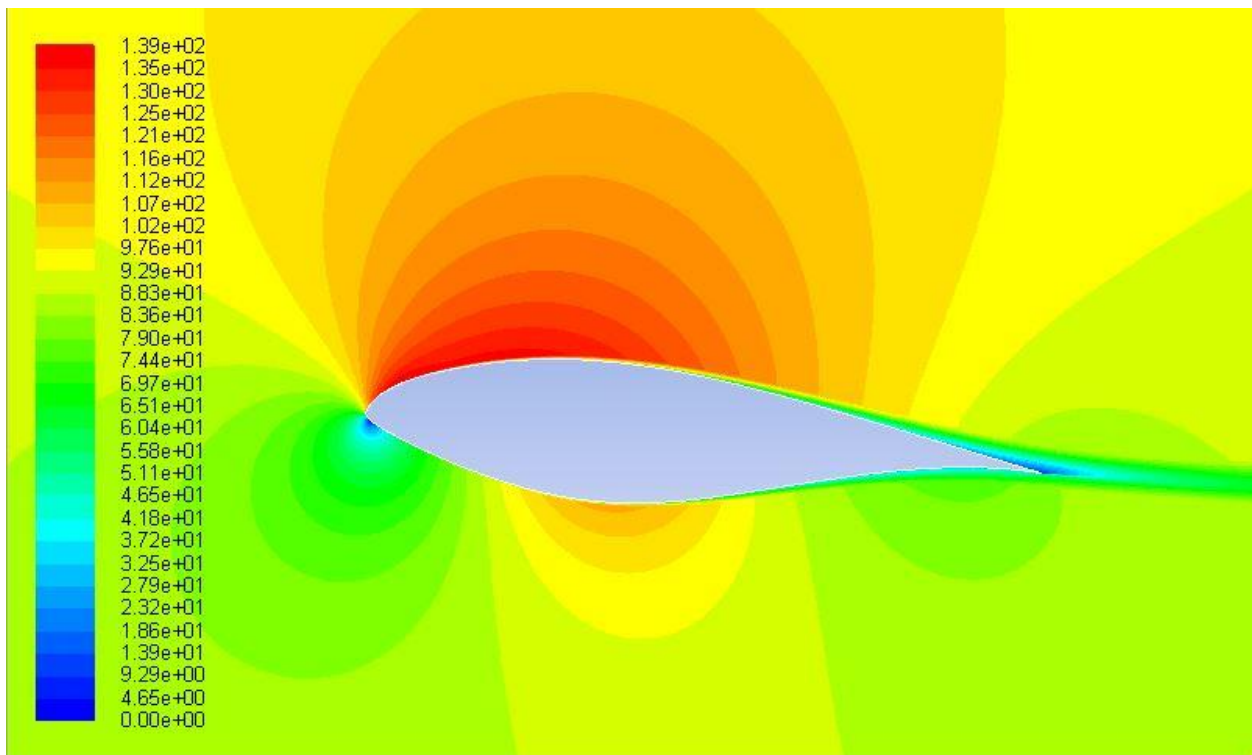


Figure 23(b) Velocity contours around optimized S809 airfoil at 58% span location

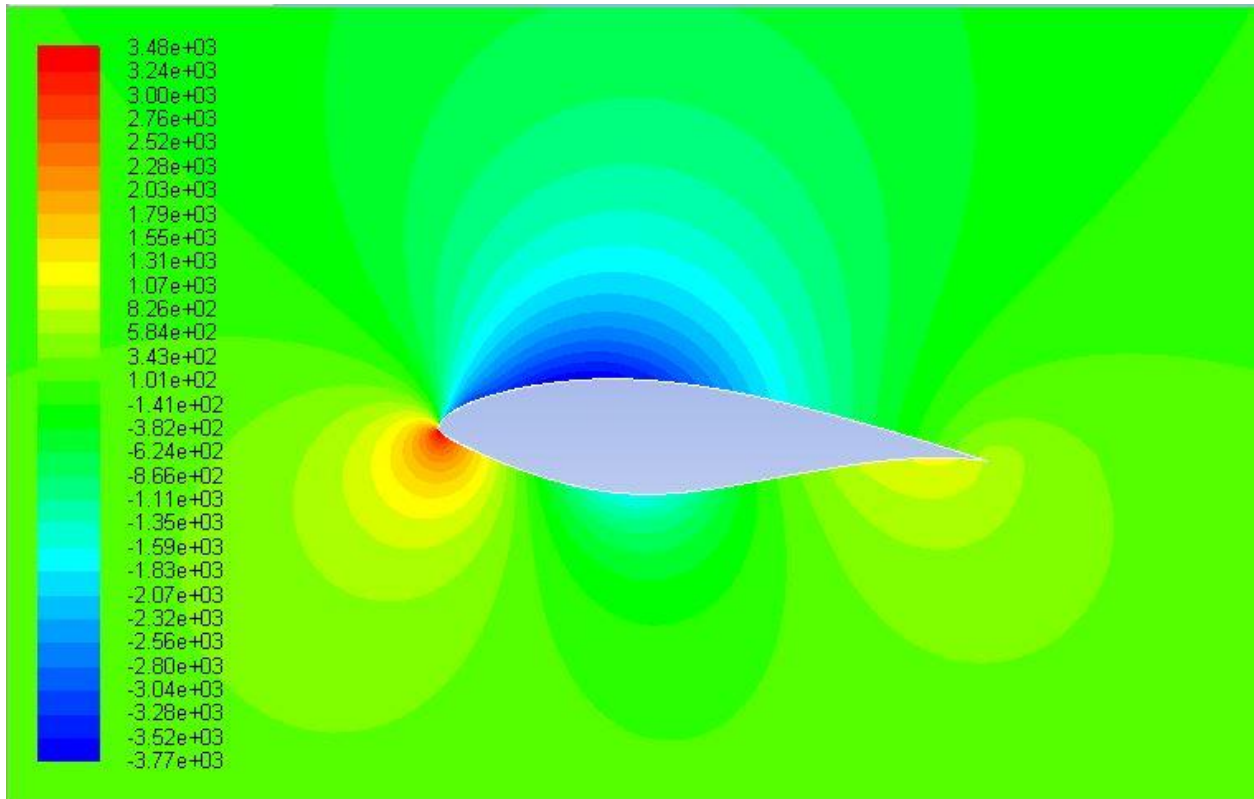


Figure 24(a) Pressure contours around optimized S809 airfoil at 98% span location

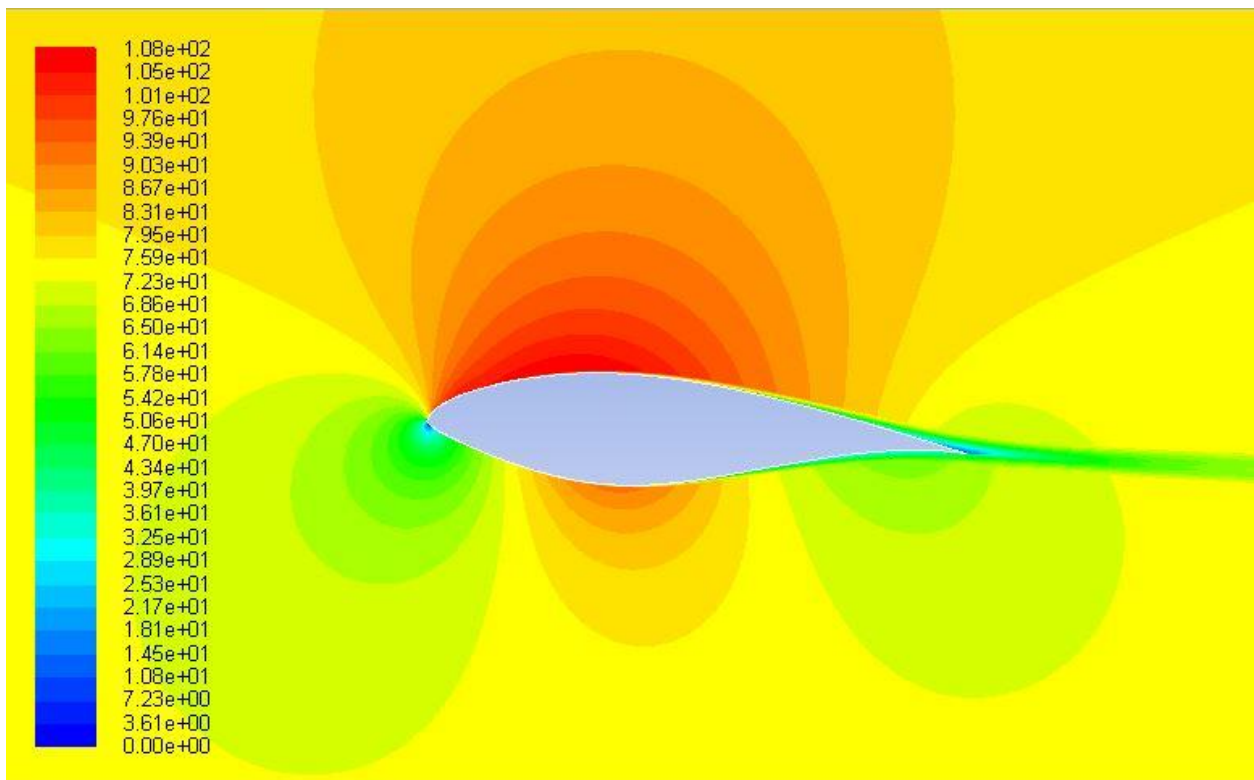


Figure 24(b) Velocity contours around optimized S809 airfoil at 98% span location

Table 6 Comparison of Cl of optimized airfoils: MOGA results and those obtained by the Adjoint Method [6]

	MOGA results at 23% Span	Adjoint Method results at 23% Span	MOGA results at 58% Span	Adjoint Method results at 58% Span	MOGA results at 98% Span	Adjoint Method results on 98% Span
Original Cl	0.64528	0.64	0.68023	0.675	0.492	0.49
Optimal Cl	0.7461	0.76	0.78274	0.78	0.63	0.68
Cl improved by	15.62%	18.75%	15.06%	15.55%	28.04%	38.78%

Table 7 Comparison of Cl/Cd of optimized airfoils: MOGA results and those obtained by the Adjoint Method [6]

	MOGA results at 23% Span	Adjoint Method results at 23% Span	MOGA results at 58% Span	Adjoint Method results at 58% Span	MOGA results at 98% Span	Adjoint Method results at 98% Span
Original Cl/Cd	40.17289	45.5	43.22983	47	37.77	36.2
Optimal Cl/Cd	48.48394	50.8	49.01435	51.8	48.89	47
Cl/Cd improved by	20.688%	13%	13.38%	10%	29.44%	29.83%

3.2 Results for optimized NACA 4412 airfoil

For NACA 4412 airfoil in ground effect, the typical flow field condition is the free stream velocity $V_\infty = 30.8\text{m/s}$ with corresponding Reynolds number based on chord length of $Re = 3 \times 10^5$. In flow field simulation, the Cartesian frame of reference is fixed to the initial position of the airfoil. In the flow field computation, compressible Reynolds-Averaged Navier-Stokes (RANS) equations are solved because of variation in density between the bottom surface of airfoil and the ground. Spalart-Allmaras (S-A) turbulence model is chosen in the calculation. The equations are solved using the finite-volume solver in ANSYS FLUENT; convection terms and diffusion terms are discretized with second-order upwind scheme and central difference scheme respectively. SIMPLEC method is employed to couple the velocity and pressure.

We choose ground clearance of $0.2C$ ($0.2 \times \text{Chord Length}$) and angle of attack of 4 degree in the optimization study. This is a typical flow condition for wing in ground (WIG) effect aircraft. The free stream wind velocity is taken as 30.8 m/s . Similar to the previous case of S809 airfoil in section 3.1, we set two optimization objectives, that is to minimize $10/C_l$ and $100 \times C_d/C_l$. The airfoil shape that results from the lowest values of $100 \times C_d/C_l$ gives the shape of the optimized airfoil. Fig. 25 shows the shape comparison of optimized NACA 4412 airfoil with ground clearance of $0.2C$ with the original airfoil. The flow field of optimized NACA 4412 airfoil with $0.2C$ ground clearance is simulated under three ground clearance of $0.4C$, $0.6C$ and $0.8C$, the lift coefficient and the lift to drag ratio for these three cases is compared in Fig. 26. The results of Fig. 26 show that the optimized airfoil with $0.2C$ ground clearance has a better performance than the original airfoil for both the aerodynamic properties. Fig. 27, 28, 29 and 30 show the comparison of pressure coefficient, pressure contours and velocity contours between the optimized airfoil (with $0.2C$ ground clearance) and the original NACA 4412 airfoil at ground heights of $0.2C$, $0.4C$, $0.6C$ and $0.8C$ respectively.

To prove the necessity of optimizing an airfoil for wing in ground effect (WIG) aircraft, we optimized the NACA 4412 airfoil in free stream. Figs. 31 and 32 respectively show the shape comparison of the optimal airfoil in free stream with the original airfoil and with the optimal airfoil for $0.2C$ ground clearance. The flow field of optimal airfoil for free stream condition is simulated under four ground clearances of $0.2C$, $0.4C$, $0.6C$ and $0.8C$. The lift coefficient and lift to drag ratio for these four cases

are compared with those of the optimal airfoil with 0.2C ground clearance. Fig. 33 and Table 8 show that the optimal airfoil for ground clearance of 0.2C has a higher lift to drag ratio than the optimal airfoil designed for free stream condition. It implies that the airfoil optimized in ground effect has a higher aerodynamic efficiency than the one optimized out of ground effect. Therefore, it is important to consider designing an airfoil specifically suited for wing in ground effect aircraft. Fig. 34, 35, 36 and 37 show the comparison of pressure coefficient, the pressure contours and the velocity contours between the optimal airfoil designed for free stream condition, the optimal airfoil for 0.2C ground clearance and the original airfoil for ground clearances of 0.2C, 0.4C, 0.6C and 0.8C respectively.

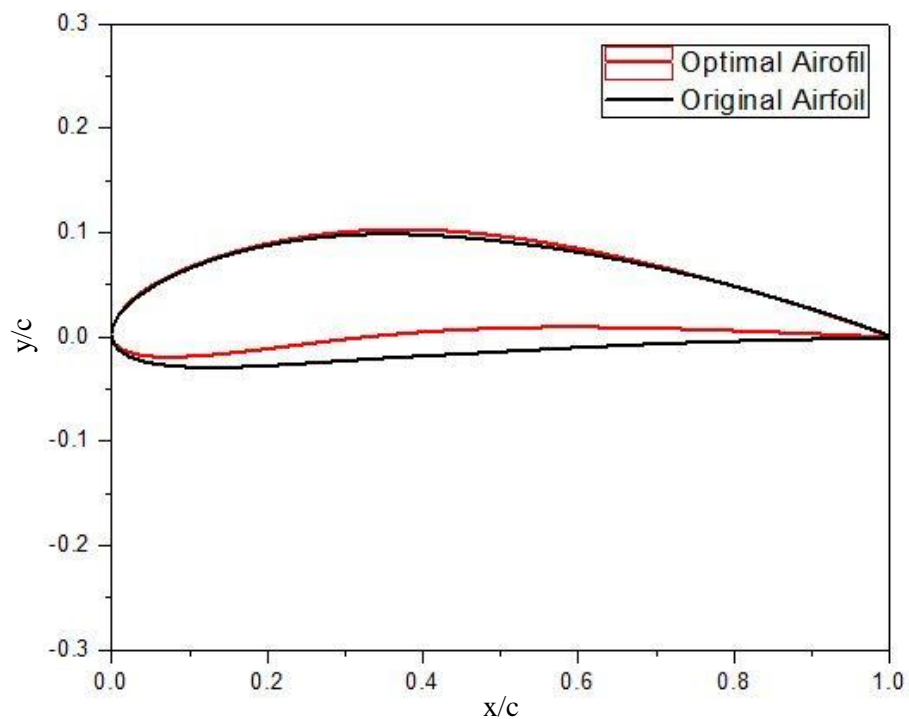


Figure 25 Comparison of shape between the optimal airfoil with 0.2C ground clearance and the original airfoil

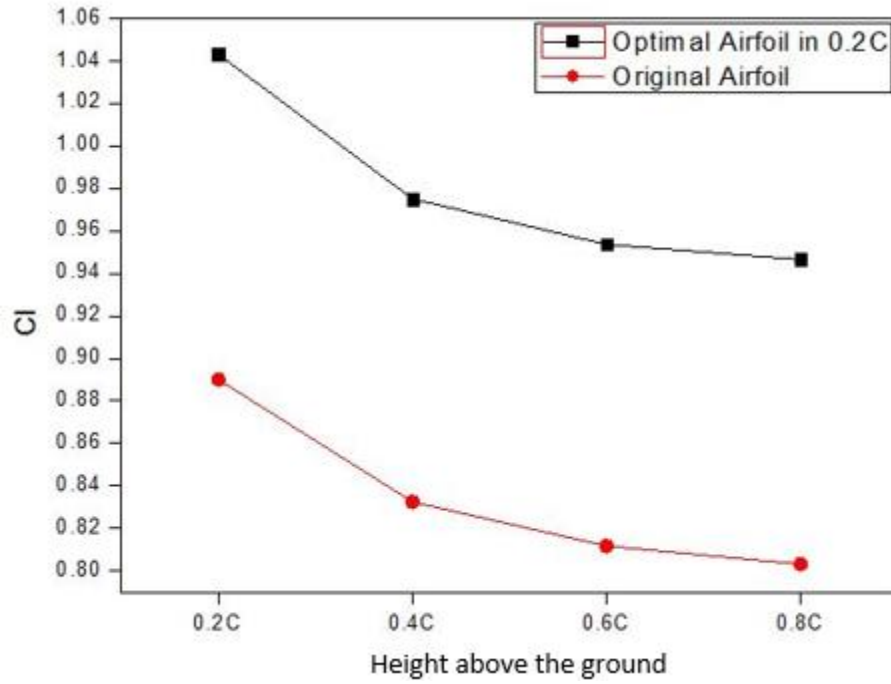


Figure 26(a) Variation of lift coefficient for the optimal airfoil with 0.2C ground clearance and the original airfoil for different ground heights

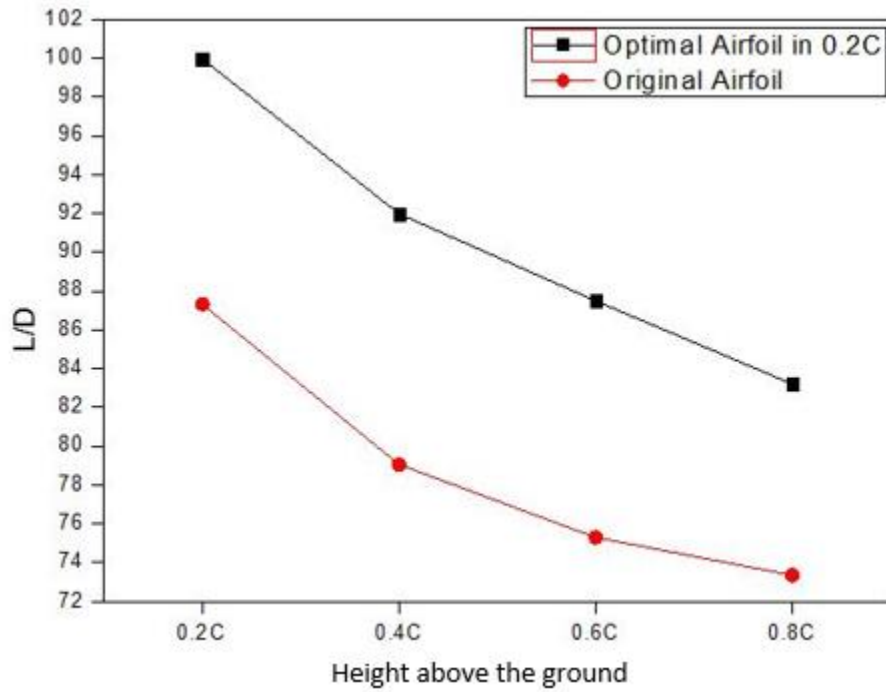


Figure 26(b) Variation of lift to drag ratio for the optimal airfoil with 0.2C ground clearance and the original airfoil for different ground heights

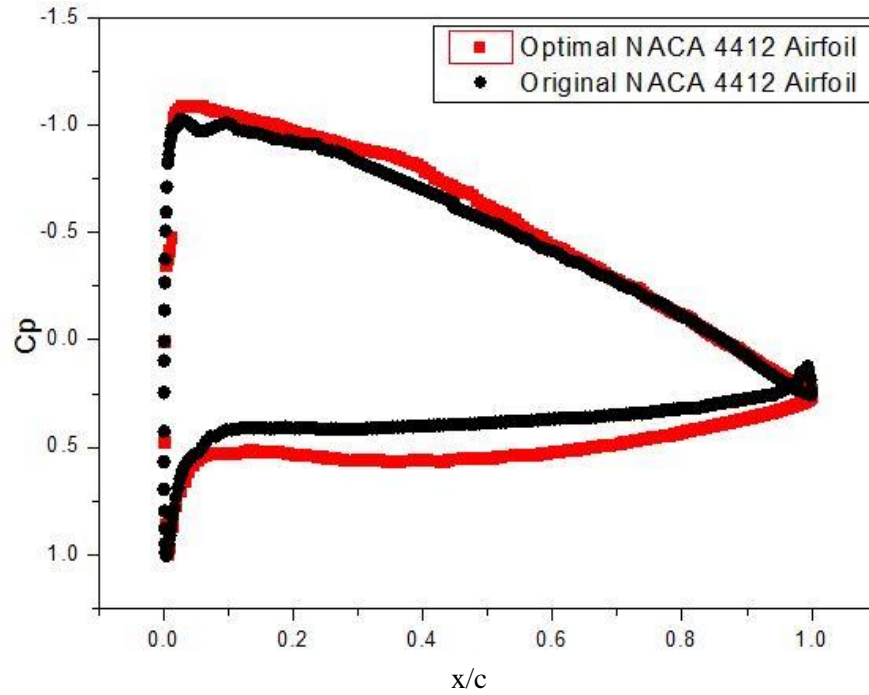


Figure 27(a) Comparison of pressure coefficient between the optimal airfoil with 0.2C ground clearance and the original airfoil

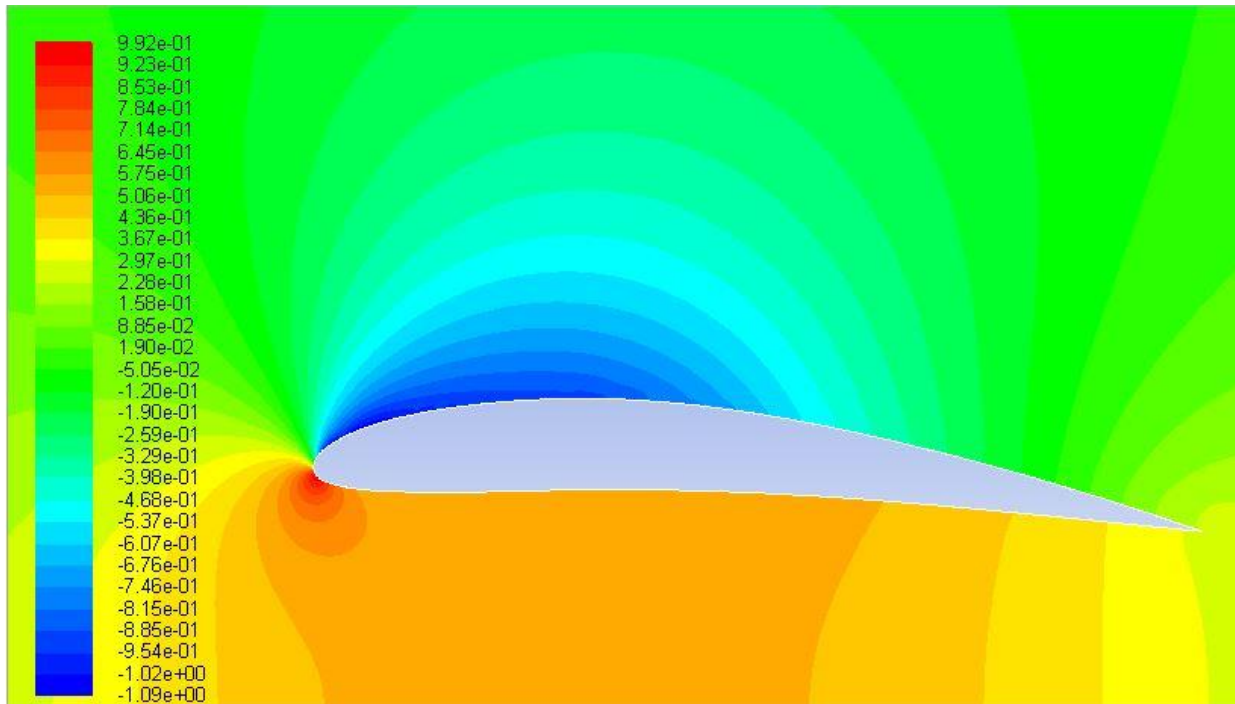


Figure 27(b) Pressure contours around the optimal airfoil with 0.2C ground clearance

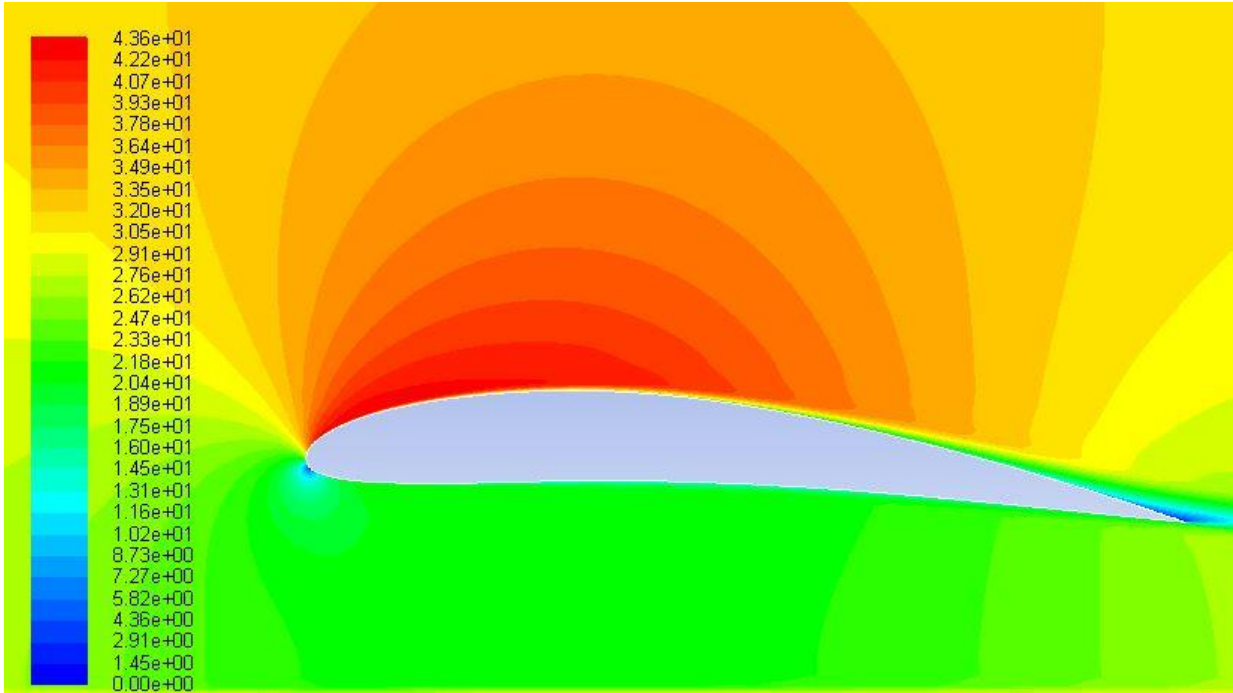


Figure 27(c) Velocity contours around the optimal airfoil with 0.2C ground clearance

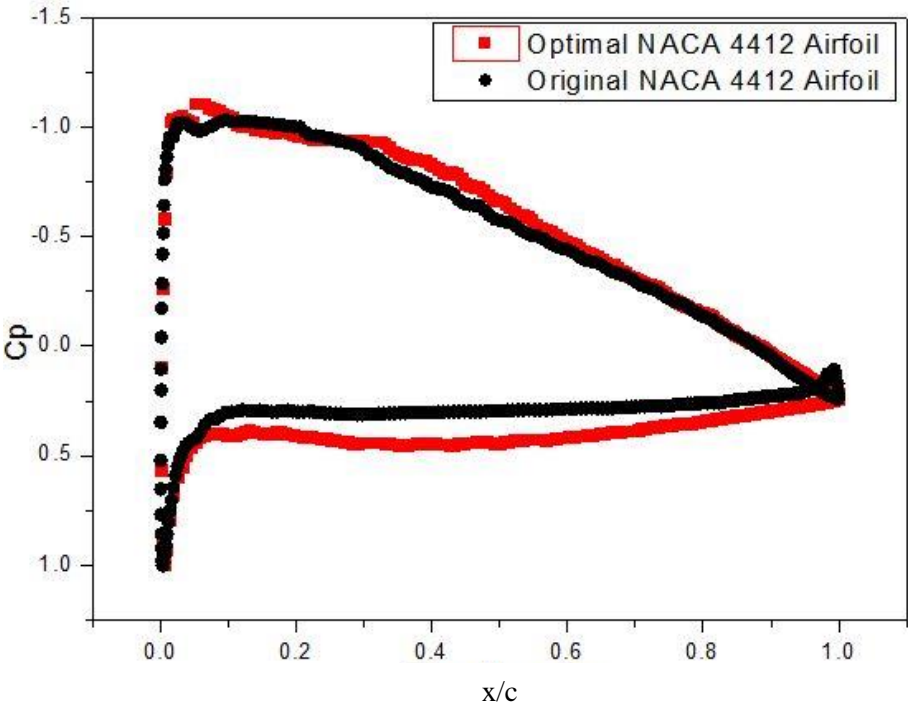


Figure 28(a) Comparison of pressure coefficient between the optimal airfoil with ground clearance of 0.2C and the original airfoil at ground height of 0.4C

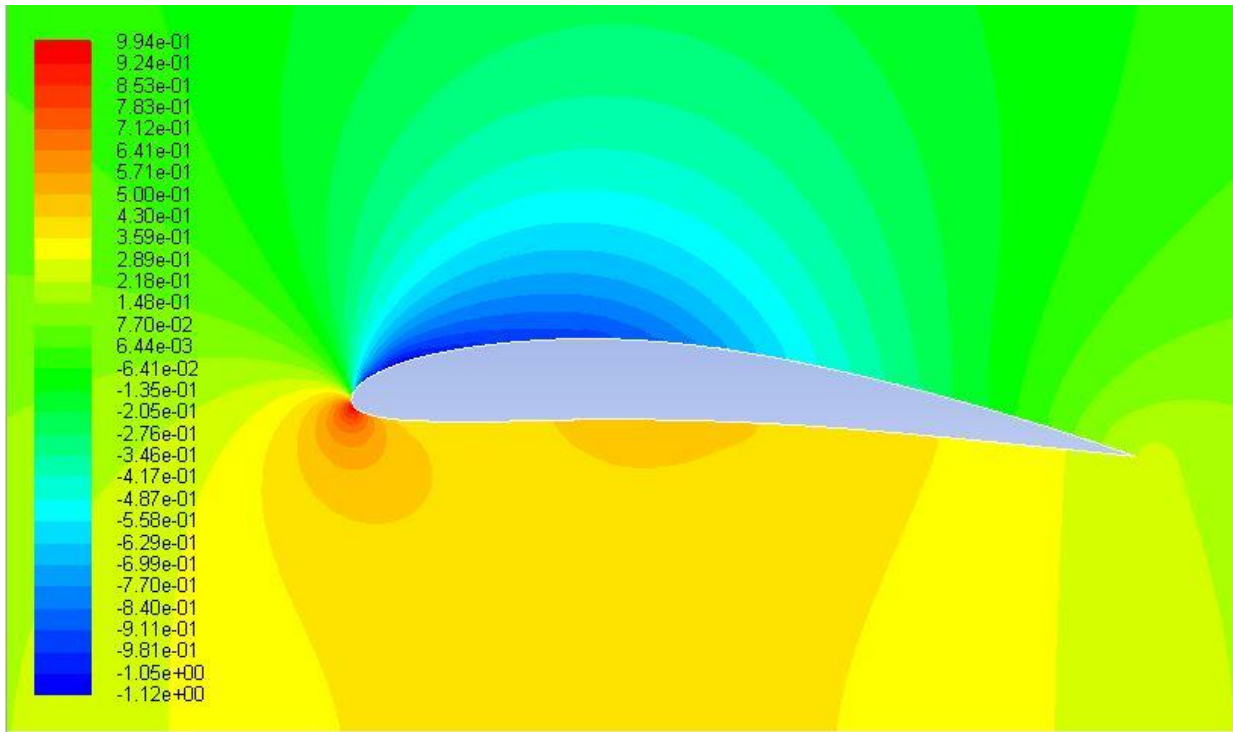


Figure 28(b) Pressure contours around the airfoil optimized for ground clearance of $0.2C$ at ground height of $0.4C$

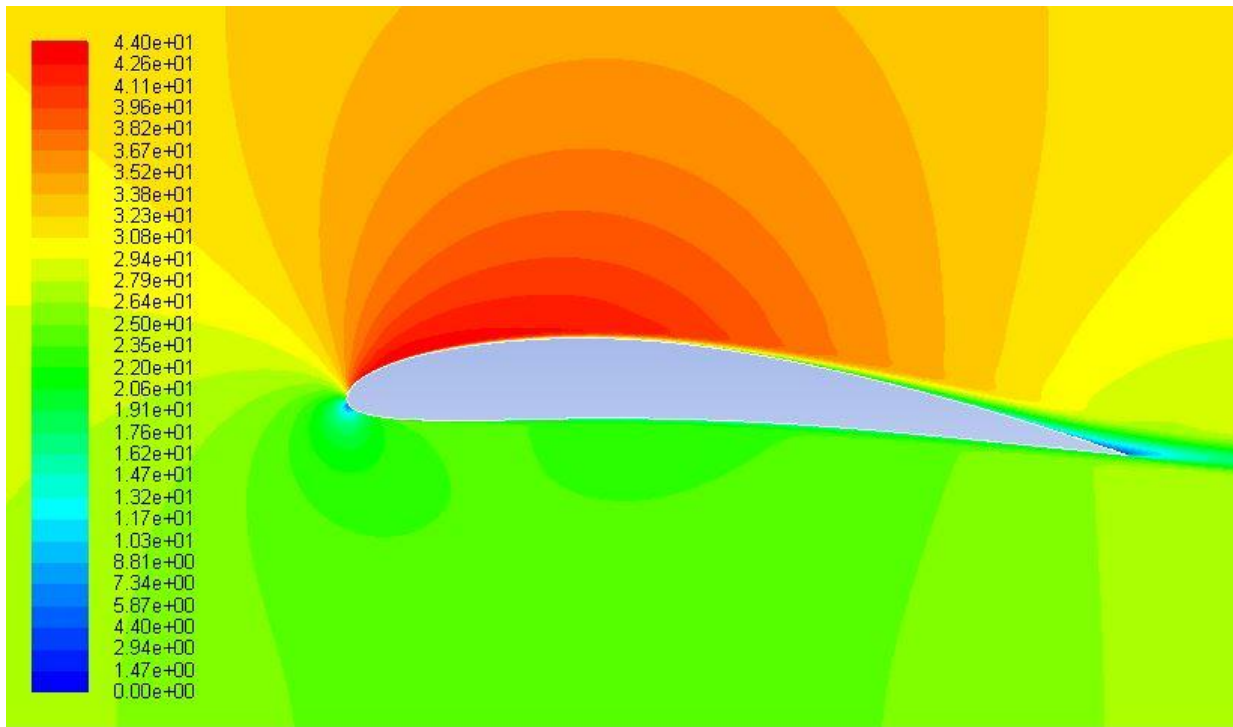


Figure 28(c) Velocity contours around the airfoil optimized for ground clearance of $0.2C$ at ground height of $0.4C$

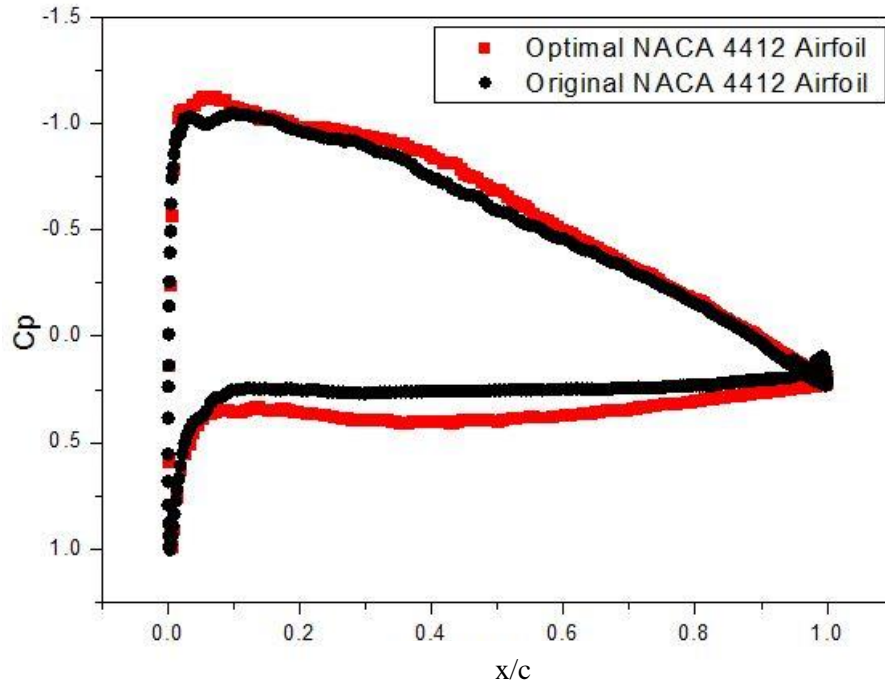


Figure 29(a) Comparison of pressure coefficient between the optimal airfoil with ground clearance of 0.2C and the original airfoil at ground height of 0.6C

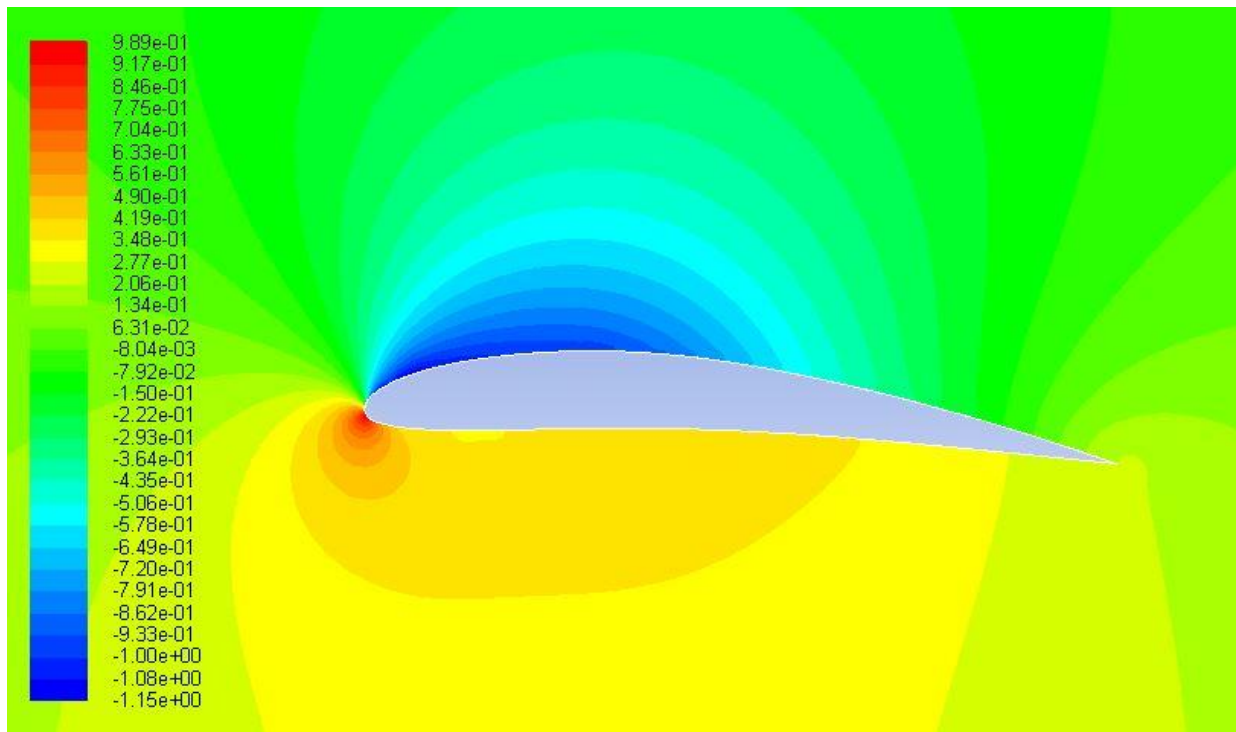


Figure 29(b) Pressure contours around the airfoil optimized for ground clearance of 0.2C at ground height of 0.6C

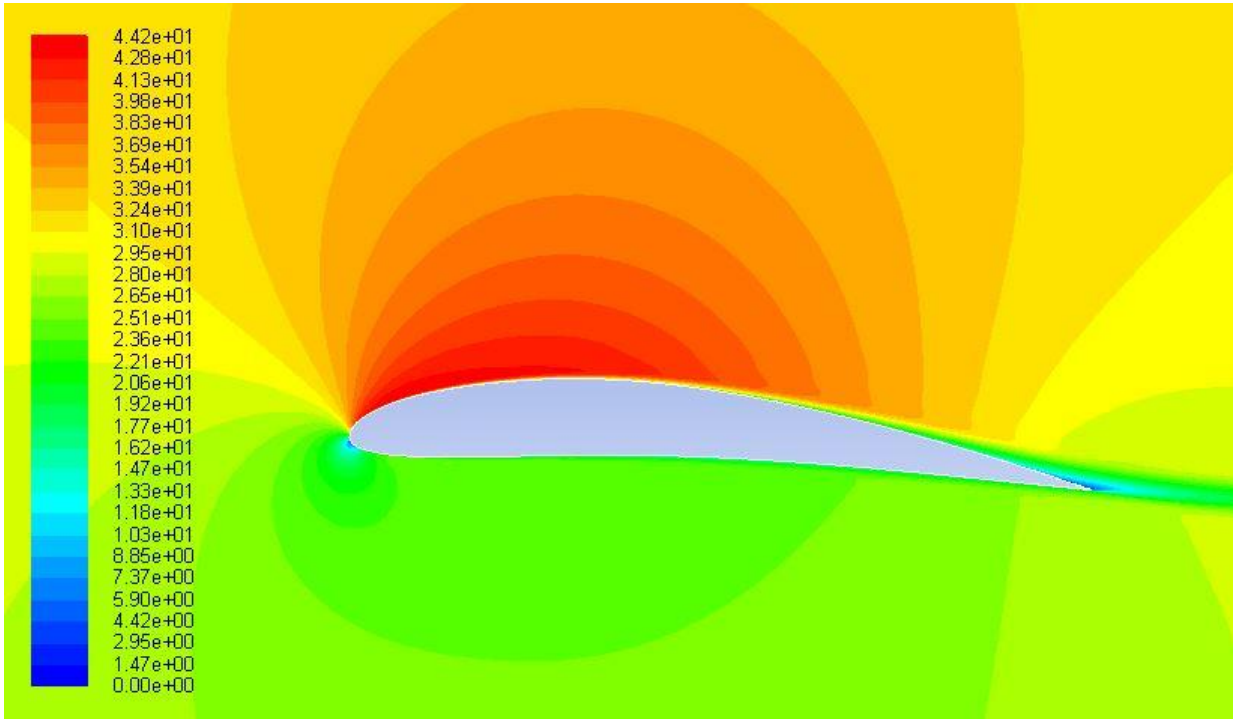


Figure 29(c) Velocity contours around the airfoil optimized for ground clearance of $0.2C$ at ground height of $0.6C$

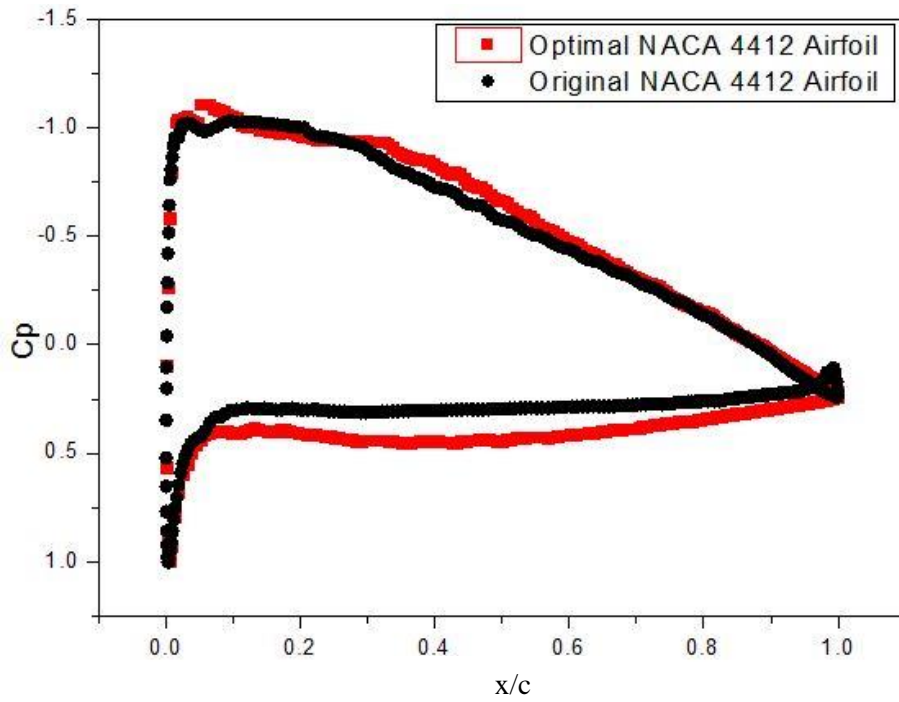


Figure 30(a) Comparison of pressure coefficient between the optimal airfoil with ground clearance of $0.2C$ and the original airfoil at ground height of $0.8C$

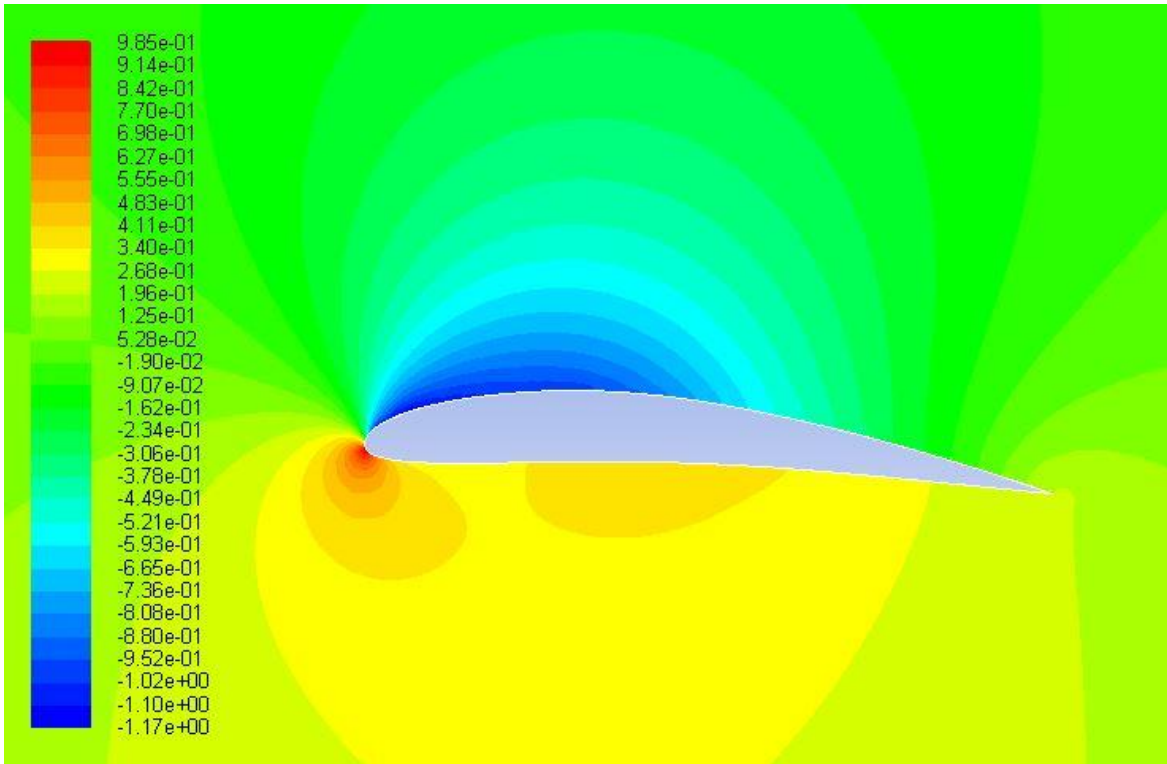


Figure 30(b) Pressure contours around the airfoil optimized for ground clearance of 0.2C at ground height of 0.8C

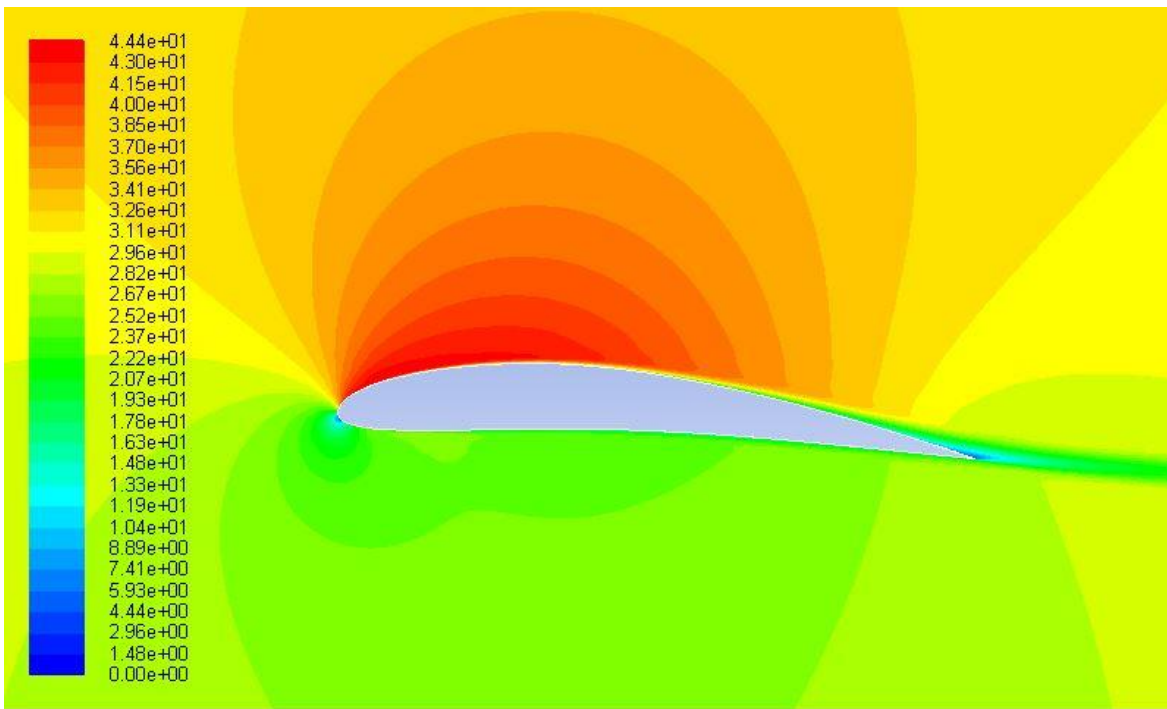


Figure 30(c) Velocity contours around the airfoil optimized for ground clearance of 0.2C at ground height of 0.8C

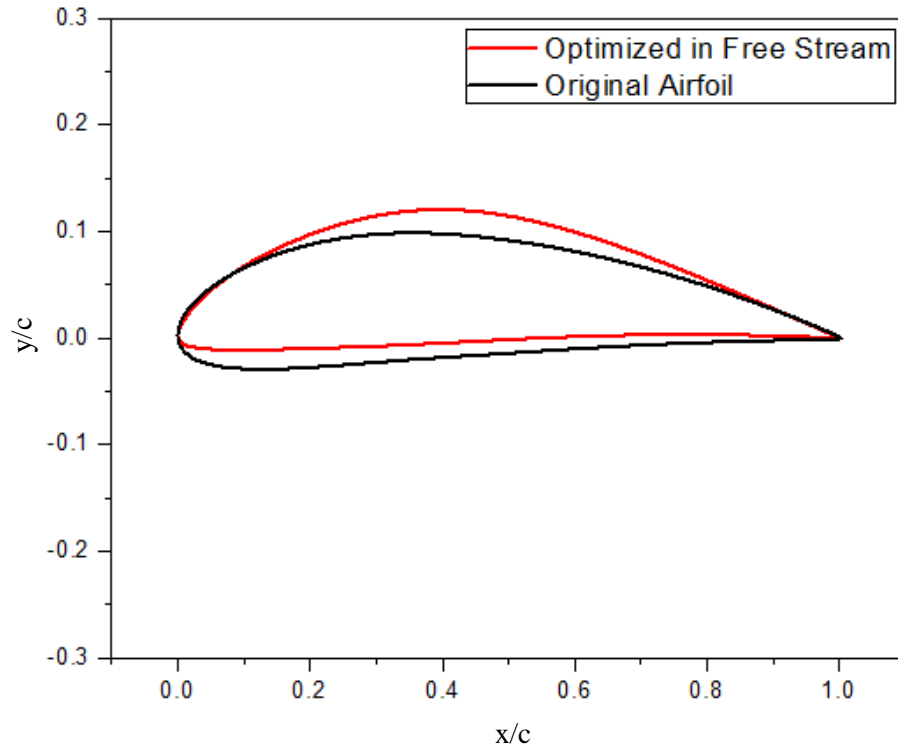


Figure 31 Comparison of shape of airfoil optimized under free stream condition (without ground effect) and the original NACA 4412 airfoil

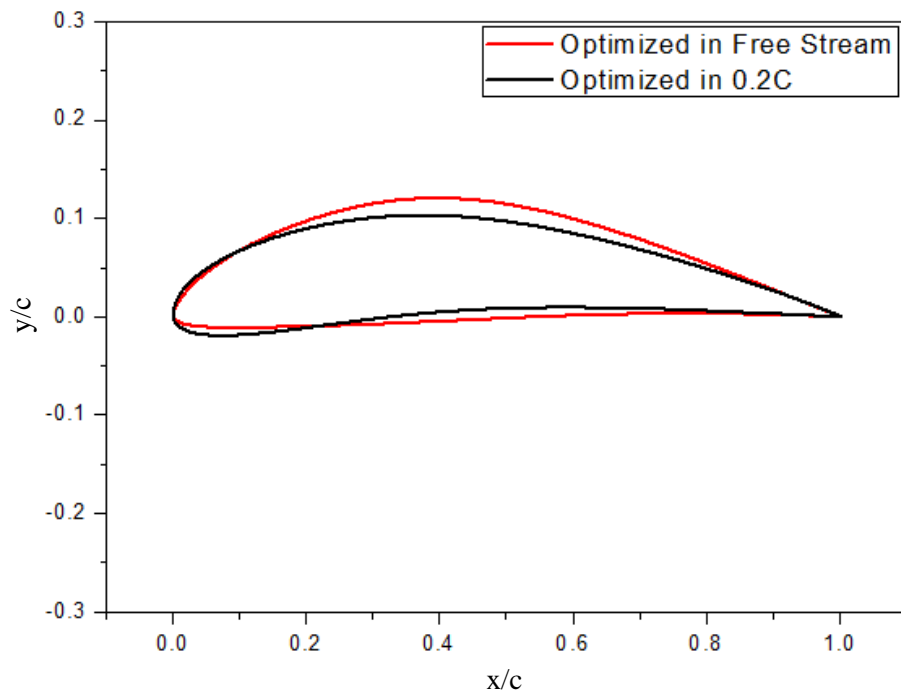


Figure 32 Comparison of shape of airfoil optimized under free stream condition (without ground effect) and the airfoil optimized in ground effect with 0.2C height above the ground

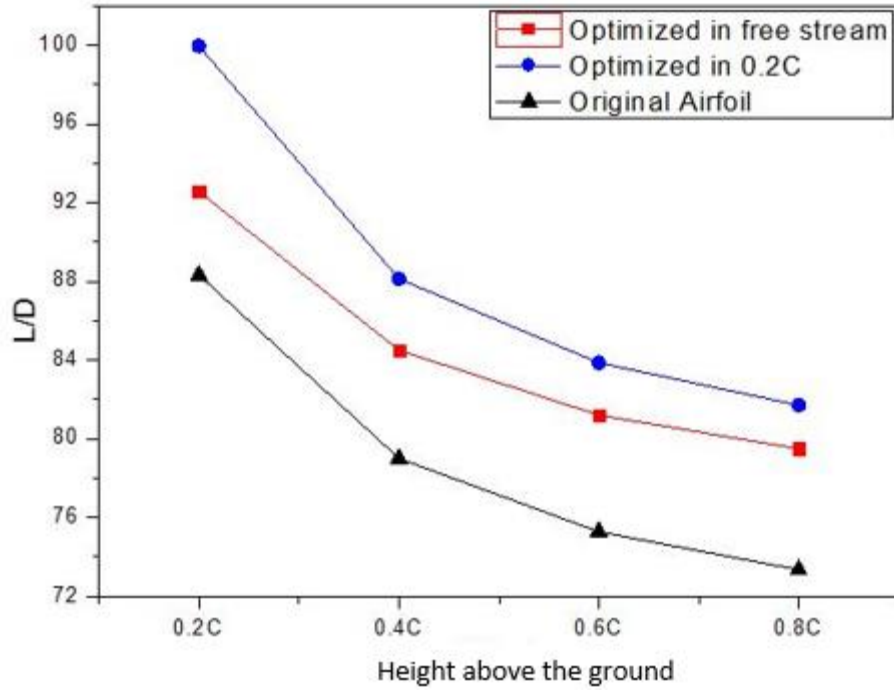


Figure 33 Comparison of lift to drag ratio between the airfoil optimized in free stream (without ground effect), the airfoil optimized in ground effect with 0.2C height above the ground and the original NACA 4412 airfoil

Table 8 Lift to drag ratio (L/D) for different airfoils

Height above the ground	Original Airfoil (L/D)	Optimal Airfoil in ground effect with 0.2C height above the ground (L/D)	Optimal Airfoil in free stream without ground effect (L/D)
0.2C	88.32	99.96	92.56
0.4C	79.01	88.13	84.50
0.6C	75.28	83.87	81.21
0.8C	73.35	81.71	79.50

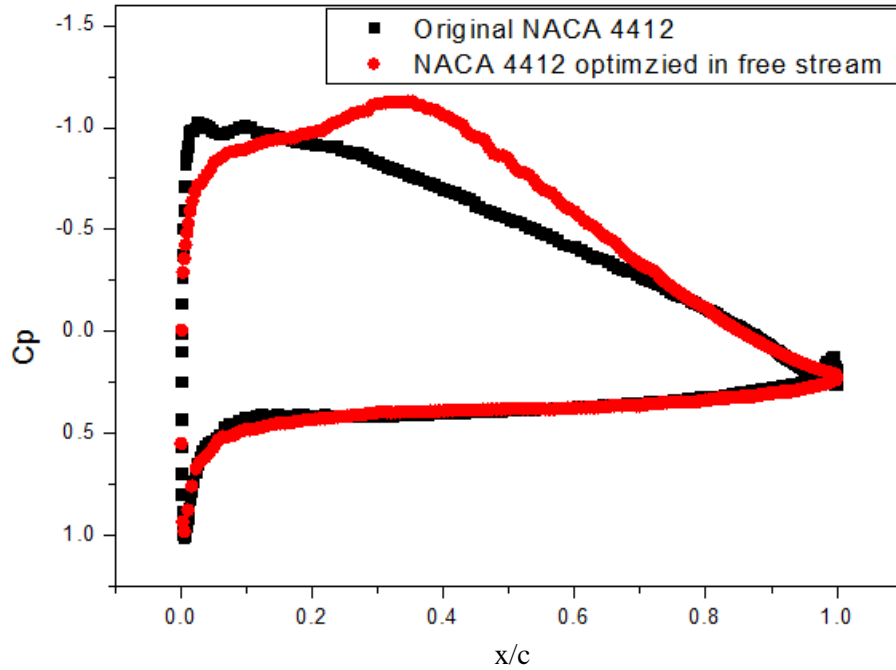


Figure 34(a) Comparison of pressure coefficient of an airfoil optimized in free stream without ground effect and the original NACA 4412 airfoil at a ground height of $0.2C$

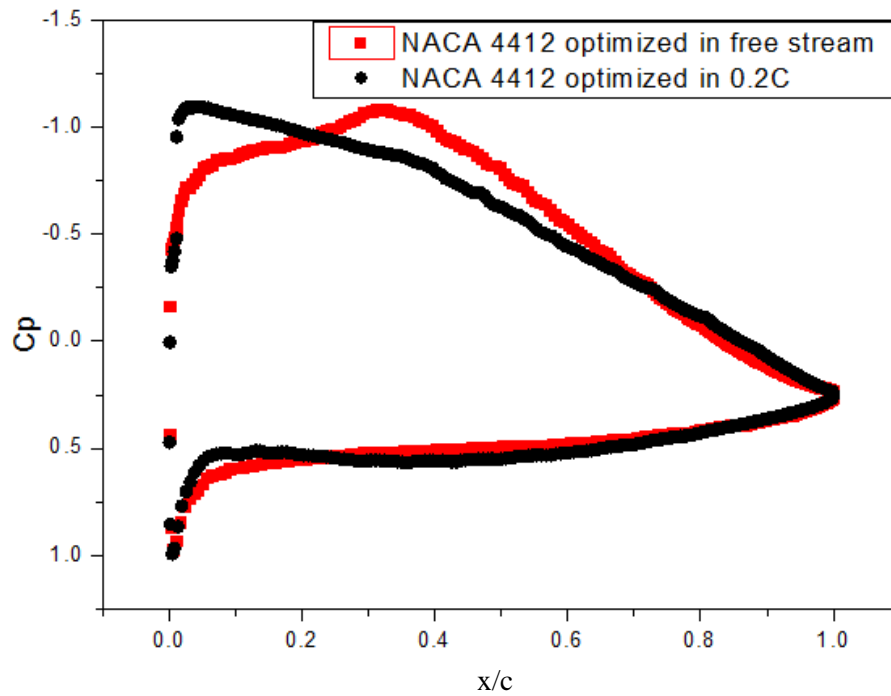


Figure 34(b) Comparison of pressure coefficient of an airfoil optimized in free stream without ground effect and the airfoil optimized in ground effect with $0.2C$ height above the ground

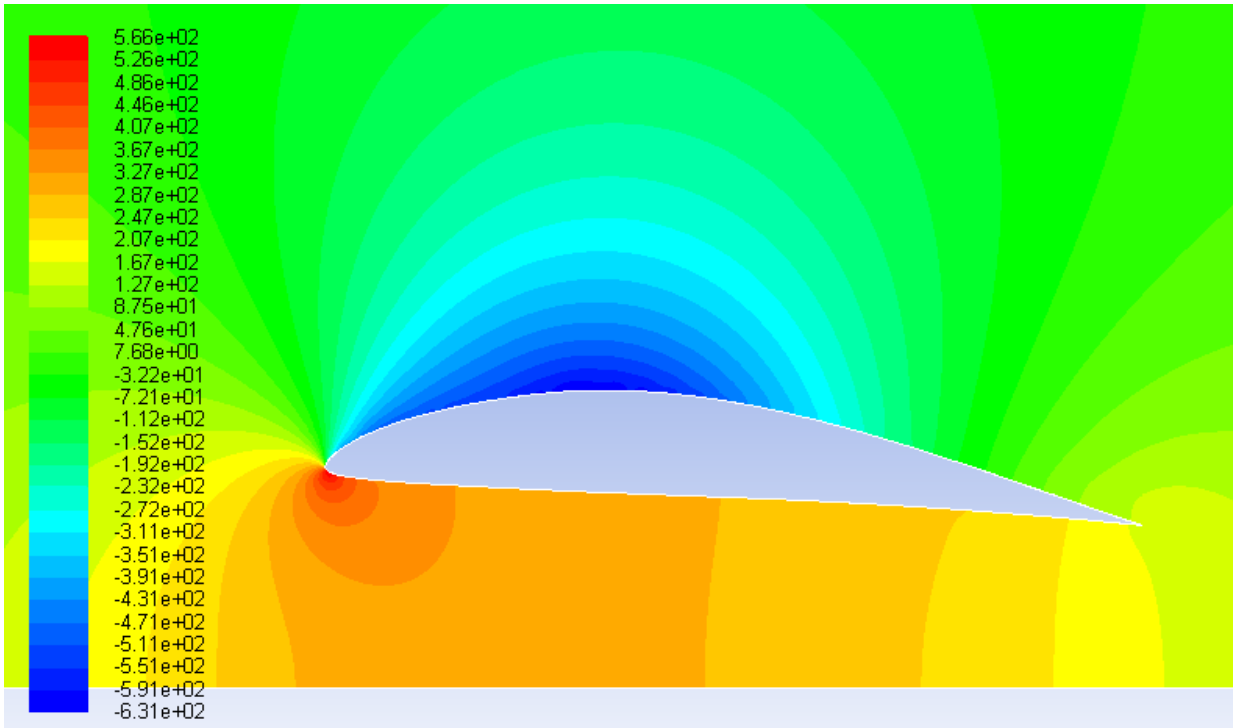


Figure 34(c) Pressure contours around the airfoil optimized in free stream without ground effect at a ground height of $0.2C$

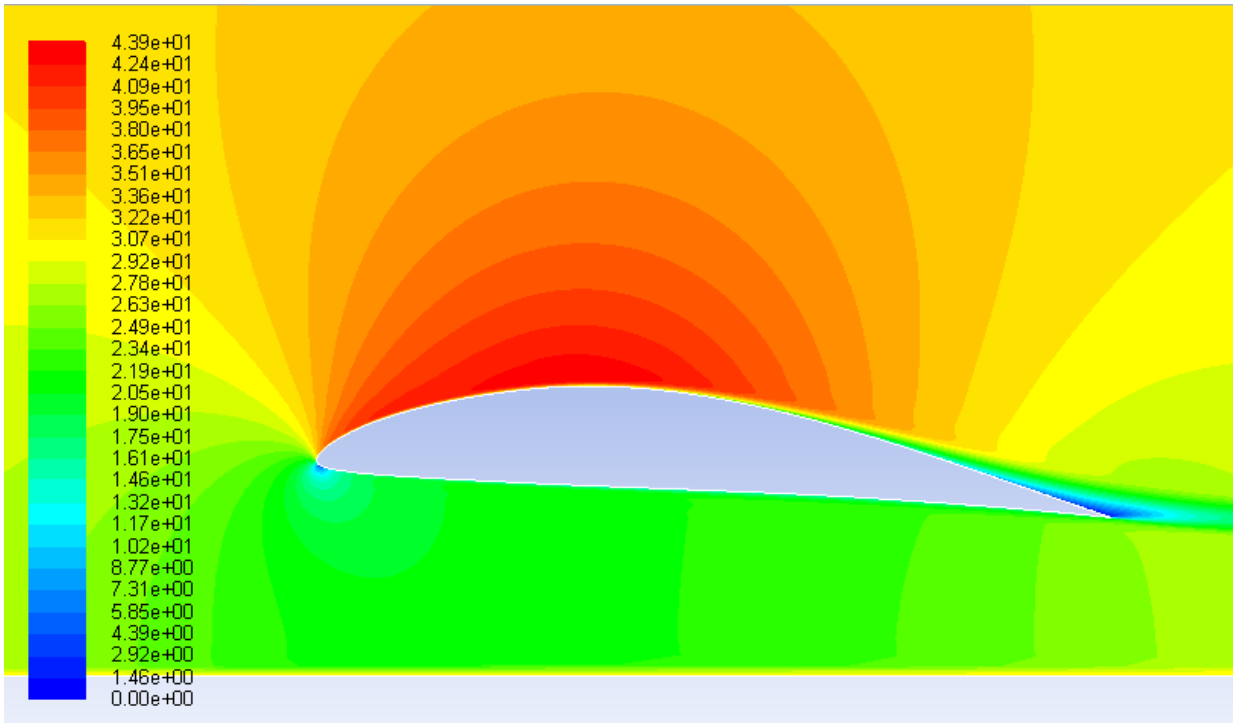


Figure 34(d) Velocity contours around the airfoil optimized in free stream without ground effect at a ground height of $0.2C$

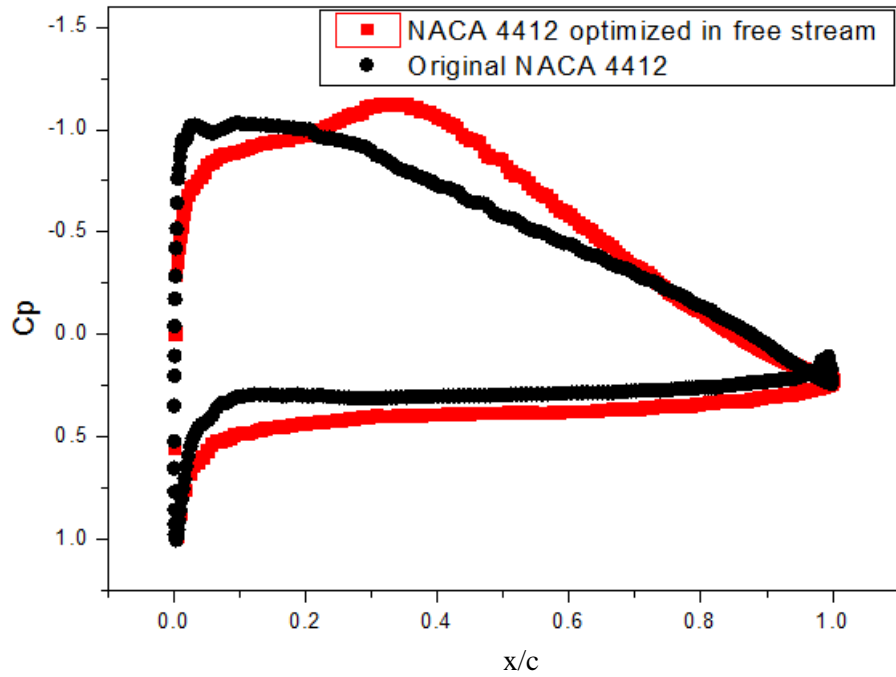


Figure 35(a) Comparison of pressure coefficient of an airfoil optimized in free stream without ground effect and the original NACA 4412 airfoil at a ground height of $0.4C$

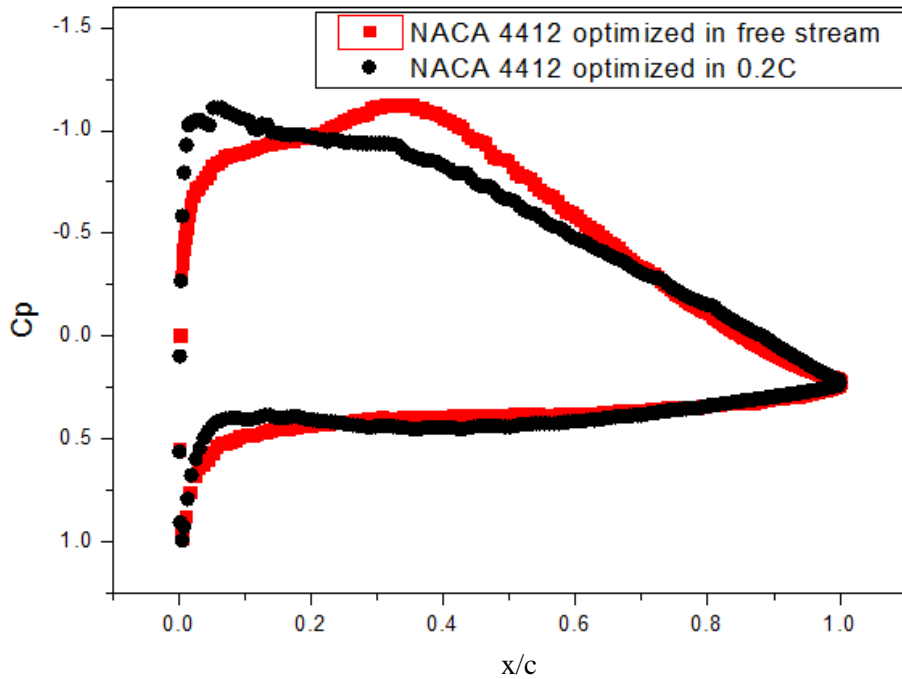


Figure 35(b) Comparison of pressure coefficient of an airfoil optimized in free stream without ground effect and the airfoil optimized in ground effect with $0.4C$ height above the ground

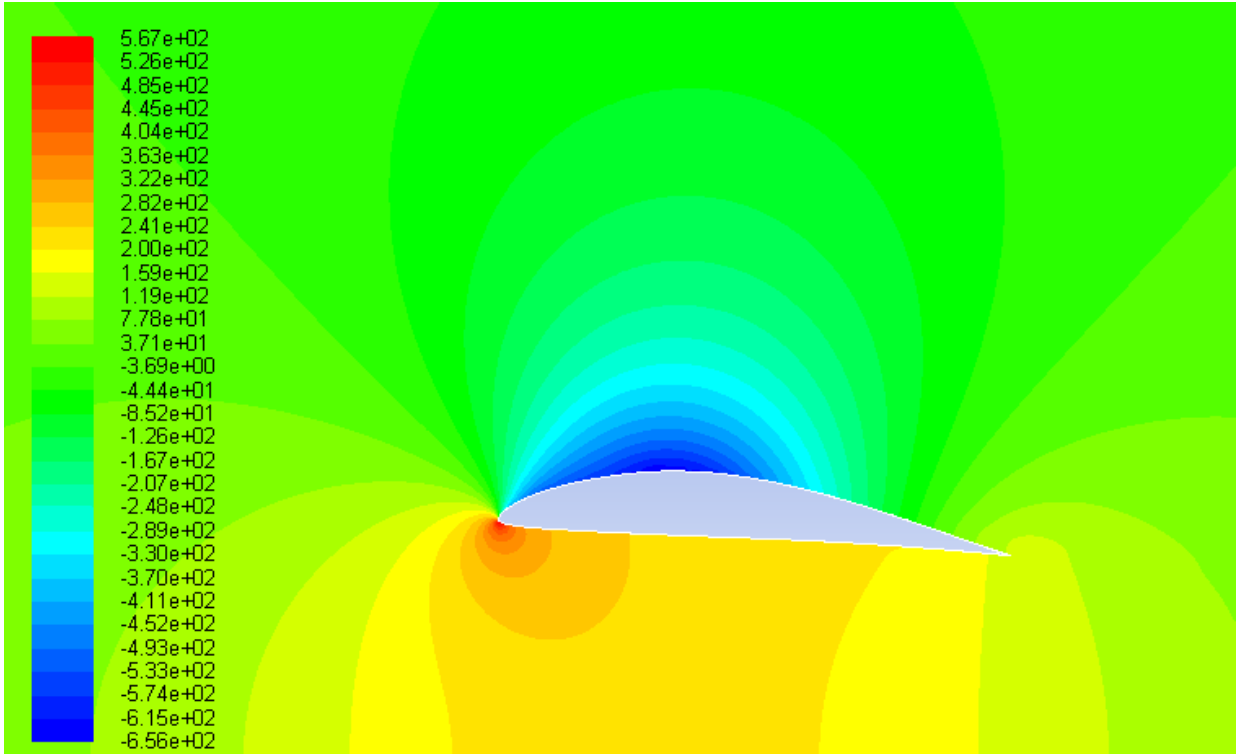


Figure 35(c) Pressure contours around the airfoil optimized in free stream without ground effect at a ground height of $0.4C$

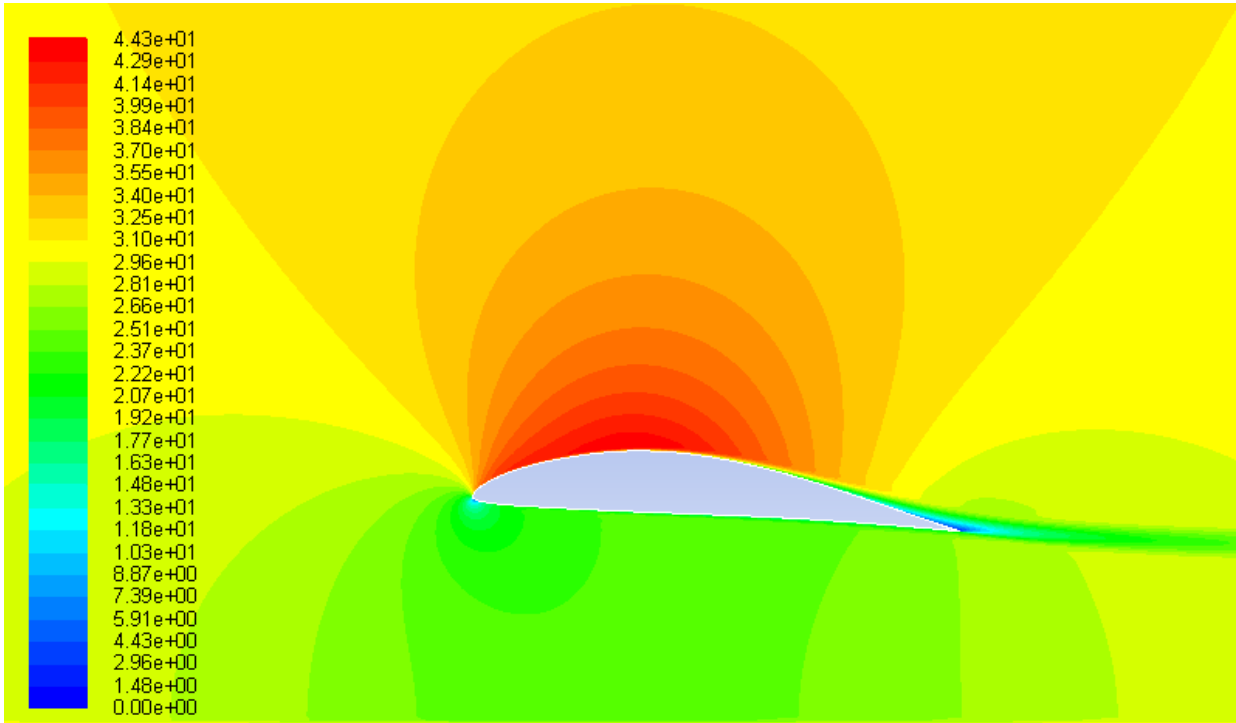


Figure 35(d) Velocity contours around the airfoil optimized in free stream without ground effect at a ground height of $0.4C$

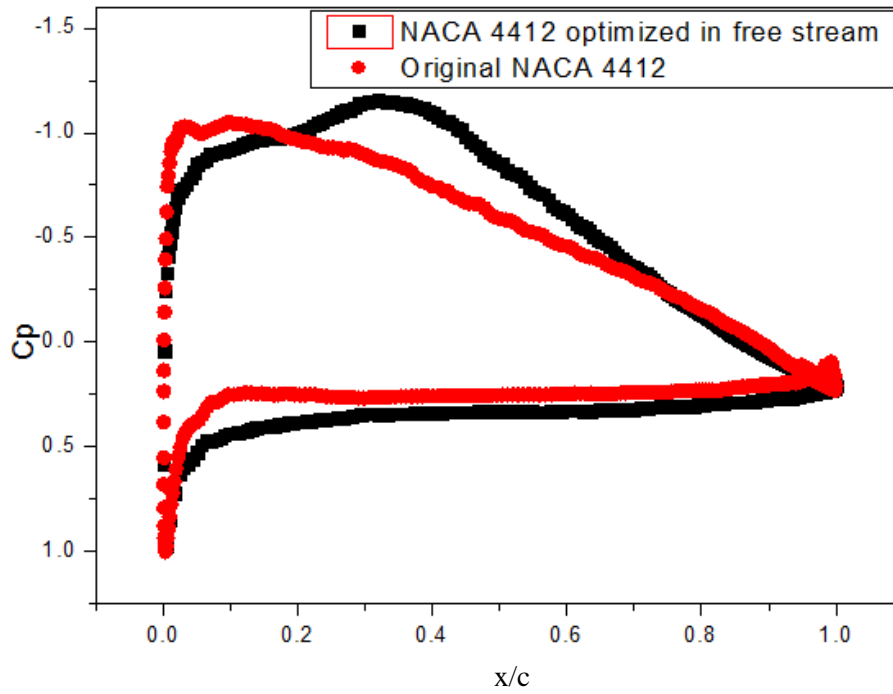


Figure 36(a) Comparison of pressure coefficient of an airfoil optimized in free stream without ground effect and the original NACA 4412 airfoil at a ground height of $0.6C$

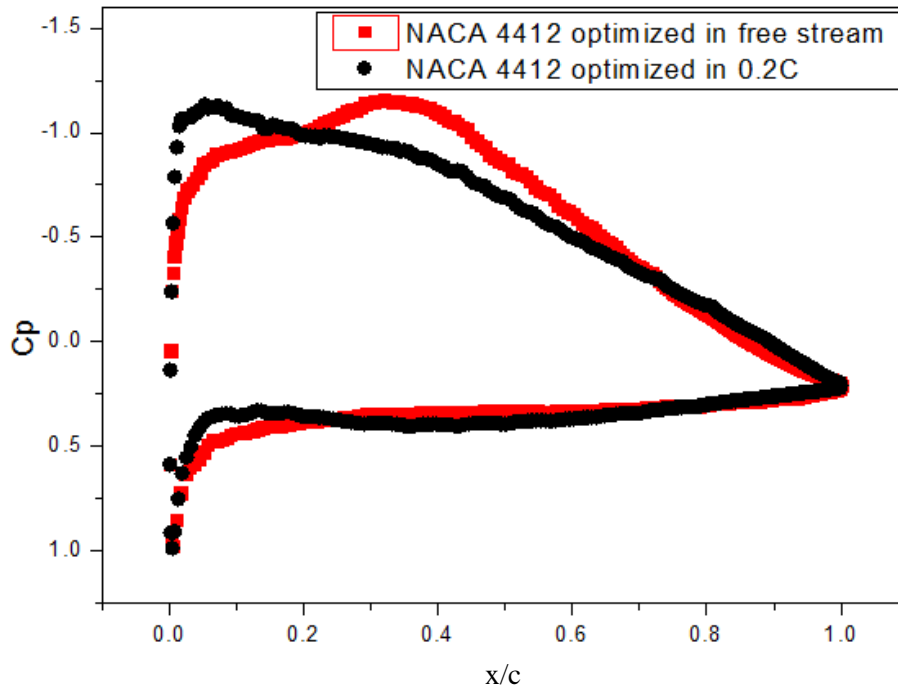


Figure 36(b) Comparison of pressure coefficient of an airfoil optimized in free stream without ground effect and the airfoil optimized in ground effect with $0.6C$ height above the ground

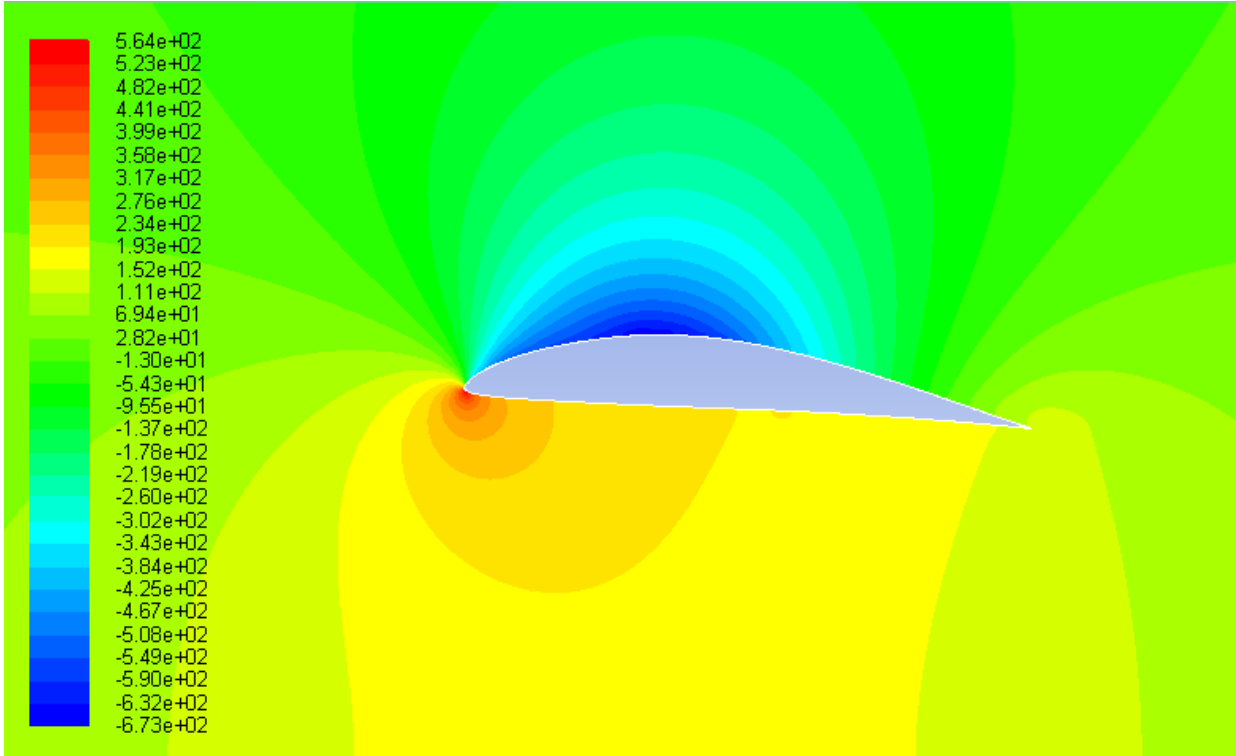


Figure 36(c) Pressure contours around the airfoil optimized in free stream without ground effect at a ground height of $0.6C$

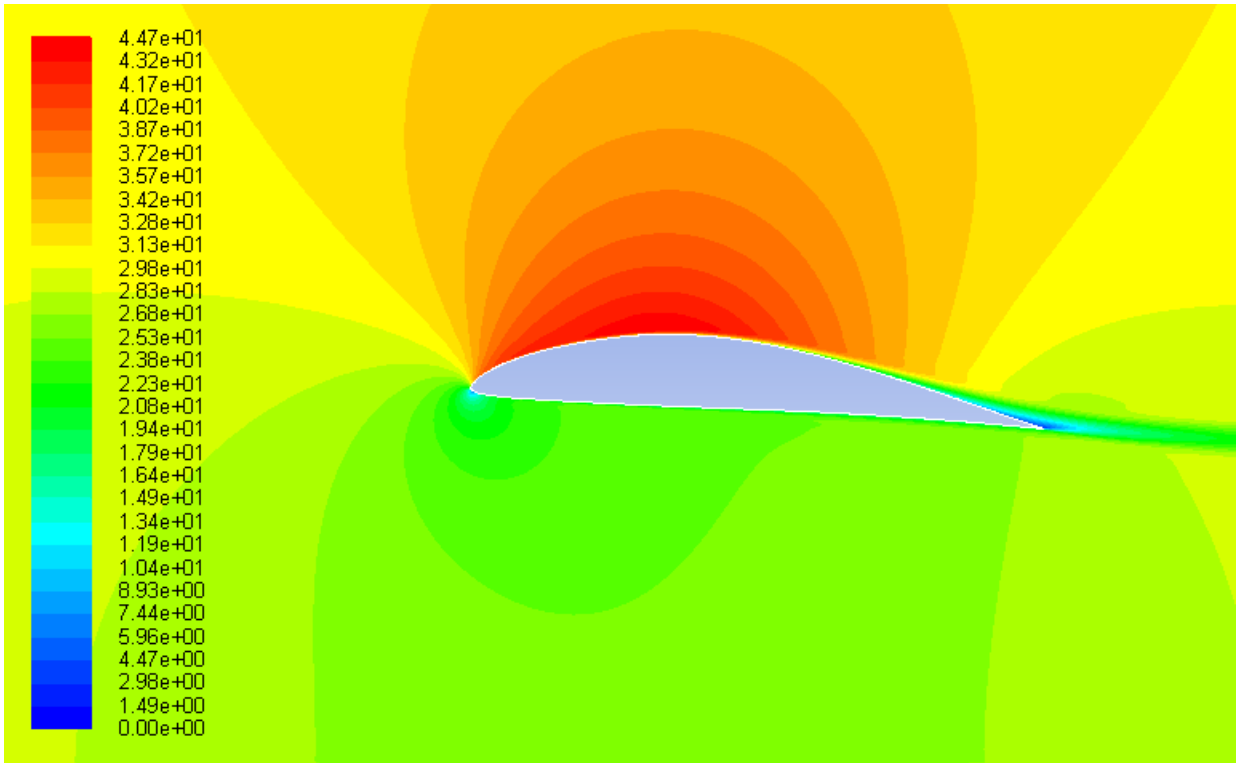


Figure 36(d) Velocity contours around the airfoil optimized in free stream without ground effect at a ground height of $0.6C$

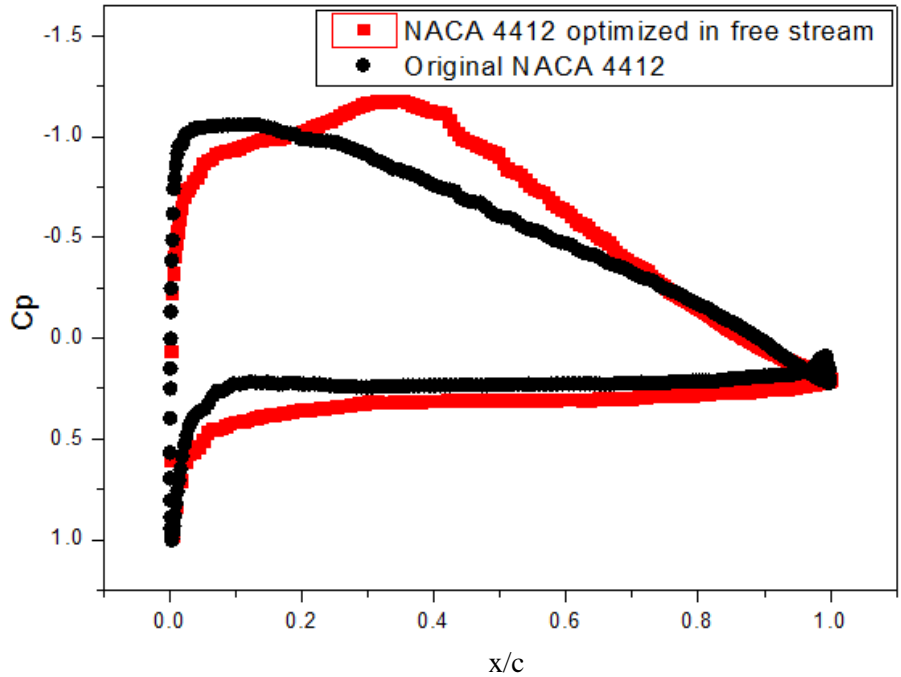


Figure 37(a) Comparison of pressure coefficient of an airfoil optimized in free stream without ground effect and the original NACA 4412 airfoil at a ground height of 0.8C

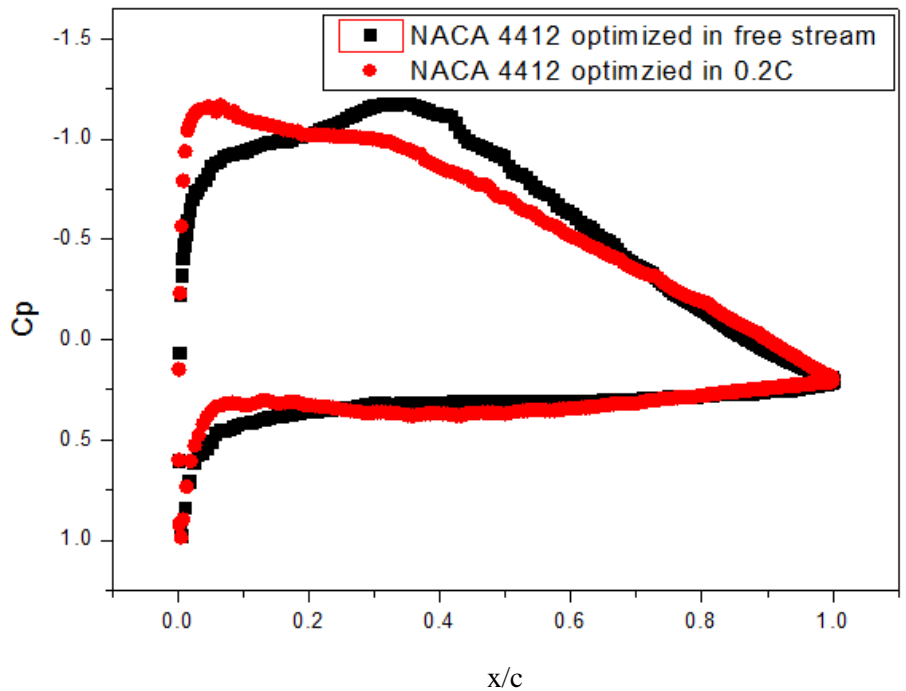


Figure 37(b) Comparison of pressure coefficient of an airfoil optimized in free stream without ground effect and the airfoil optimized in ground effect with 0.8C height above the ground

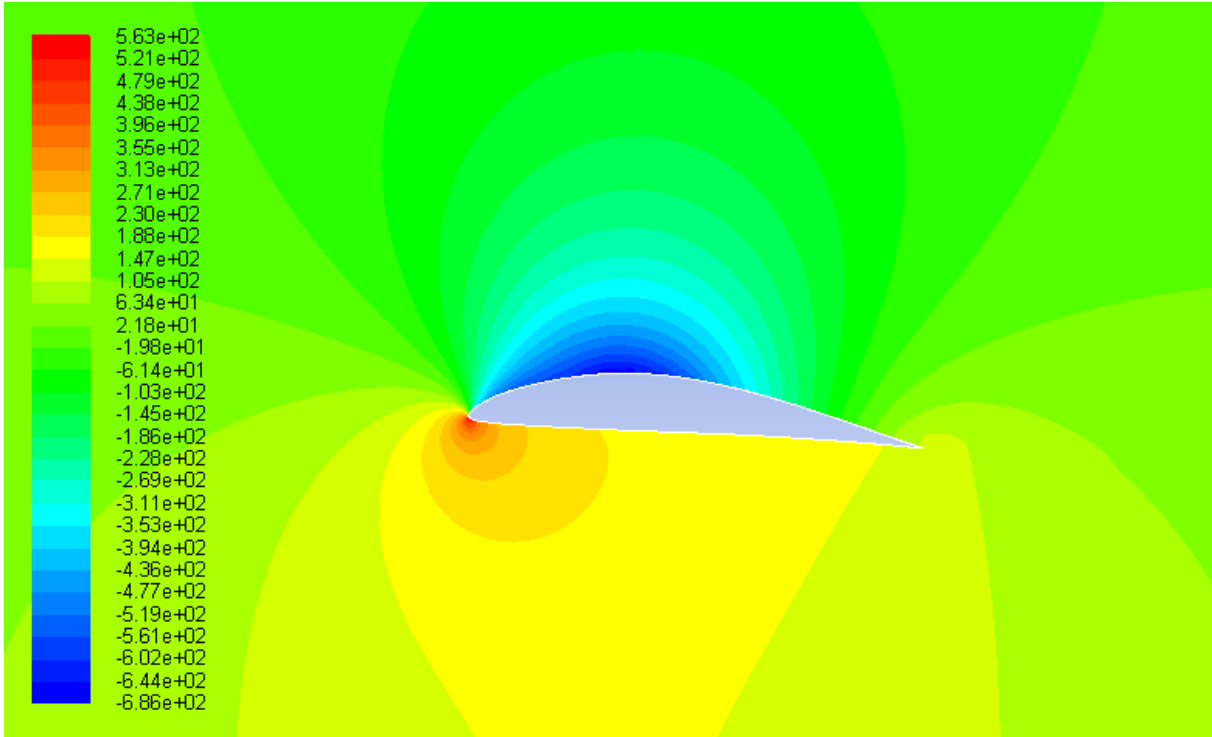


Figure 37(c) Pressure contours around the airfoil optimized in free stream without ground effect at a ground height of 0.8C

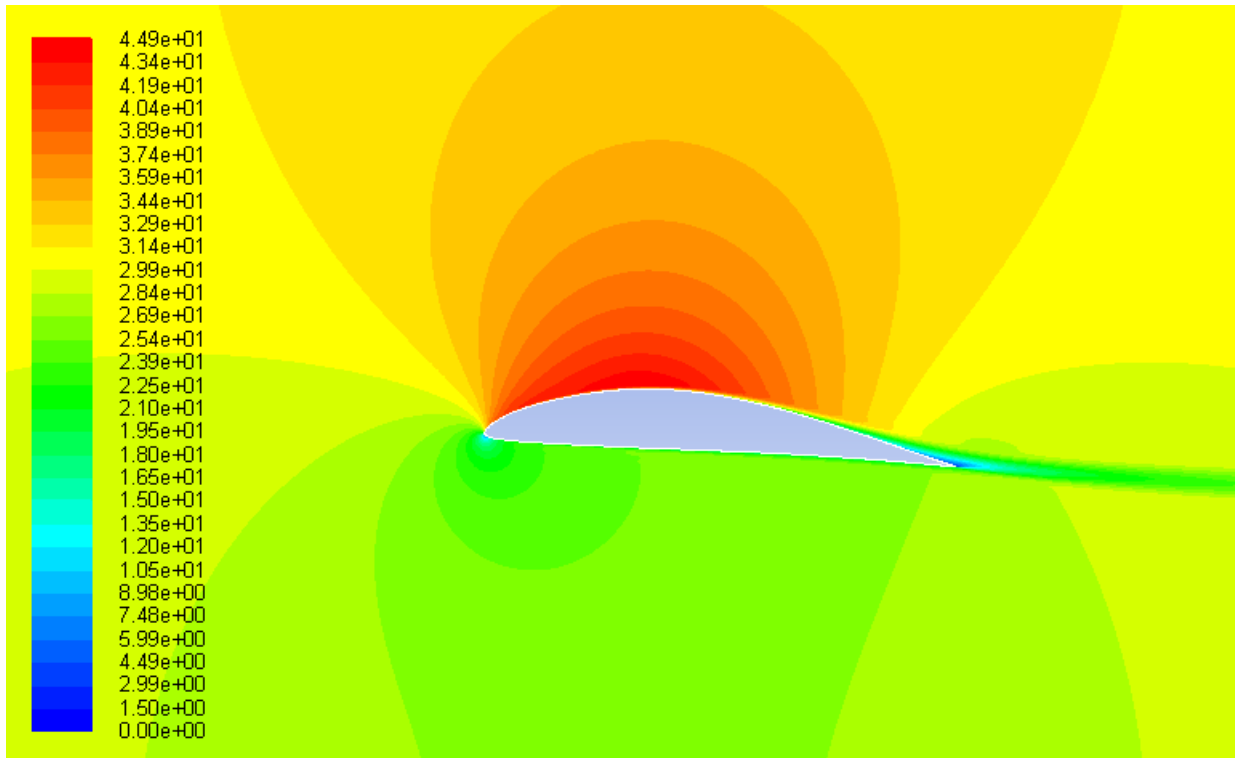


Figure 37(d) Velocity contours around the airfoil optimized in free stream without ground effect at a ground height of 0.8C

Chapter 4

Conclusions

In this thesis, we have employed a multi-objective genetic algorithm (MOGA) to optimize the shape of a well-known wind turbine airfoil S809 and wing in ground effect airfoil NACA 4412 to improve their lift and drag characteristics, in particular to achieve two objectives that is to increase the lift as well as the lift to drag ratio. The commercially available software FLUENT is employed to calculate the flow field on an adaptive structured mesh using the Reynolds-Averaged Navier-Stokes (RANS) equations. A two equation SST $k - \omega$ turbulence model is employed in calculating turbulence flow field of S809 airfoil and one equation Spalart-Allmaras (S-A) turbulence model is used in calculating the turbulent flow field of NACA 4412. The optimization results show significant improvement in both the lift coefficient and the lift to drag ratio of the optimized airfoils compared to the original airfoils. In case of S809 airfoil, MOGA results are in close agreement with those obtained by using adjoint-equation based optimization technique [6]. In case of NACA 4412, the optimization results show the importance of designing an airfoil specifically suited for wing in ground effect craft.

References

- [1] Halloran, M and Meara, S.O', "Wing in Ground Effect Craft Review," Contract Report CR-9802, The Sir Lawrence Wackett Centre for Aerospace Design Technology, Royal Melbourne Institute of Technology, Melbourne, 1999.
- [2] Grasso, F., "Usage of Numerical Optimization in Wind Turbine Airfoil Design," *Journal of Aircraft*, Vol. 48, No. 1, 2011, pp. 248-255. doi: 10.2514/1.C031089
- [3] Tangler, J. L. and Somers, D. M., "NREL Airfoil Families for HAWTs," *Wind Power 1995*, Washington D.C., 1995, pp. 117-123.
- [4] Somers, D. M., "Design and Experimental Results for the S809 Airfoil," NREL Report, 1997.
- [5] Giguère, P., "Design of a Tapered and Twisted Blade for the NREL Combined Experiment Rotor," NREL Report, 1999.
- [5] Ritlop, R. and Nadarajah, S. K., "Toward the Aerodynamic Shape Optimization of Wind Turbine Profiles," *Proc. of the 17th Annual Conference of the CFD Society of Canada*. Ottawa, Canada, 2009.
- [7] Rozhdestvensky, K., *Aerodynamics of a Lifting System in Extreme Ground Effect*, Springer, 2000
- [8] Yun, L., Bliault, A. and Doo, J., *WIG Craft and Ekranoplan: Ground Effect Craft Technology*, Springer, 2010.
- [9] Doig, G. and Barber, T. J., "Considerations for Numerical Modeling of Inverted Wings in Ground Effect," *AIAA Journal*, Vol. 49, No. 10, 2011, pp. 2330-2333. doi: Doi 10.2514/1.J051273
- [10] Ahmed, M. R., Takasaki, T. and Kohama, Y., "Aerodynamics of a NACA4412 Airfoil in Ground Effect," *AIAA Journal*, Vol. 45, No. 1, 2007, pp. 2330-2333. doi: Doi 10.2514/1.J051273
- [11] Mahon, S. and Zhang, X., "Computational Analysis of Pressure and Wake Characteristics of an Aerofoil in Ground Effect," *Journal of Fluids Engineering-Transactions of the ASME*, Vol. 127, No. 2, 2005, pp. 290-298. doi: Doi 10.1115/1.1891152
- [12] Wang, D. F. and Dai, C., "Experimental Research on the Aerodynamics of a Symmetrical Airfoil near Free Surface," *Journal of Fluids Engineering, Transactions of the ASME*, Vol.127, No.2, 2005, pp. 290-298. doi: Doi 10.1115/1.1891152
- [13] Lee, S. H. and Lee, J., "Optimization of Three-Dimensional Wings in Ground Effect Using Multiobjective Genetic Algorithm," *Journal of Aircraft*, Vol. 48, No. 5, 2011, pp. 1633-1645. doi: Doi 10.2514/1.C031328
- [14] Zhang, X. and Zerihan, J. "Edge Vortices of a Double-Element Wing in Ground Effect," *Journal of Aircraft*, Vol. 41, No. 5, 2004, pp. 1127-1137. doi: Doi 10.2514/1.1380
- [15] Yang, Z. G. and Yang, W., "Complex Flow for Wing-in-Ground Effect Craft with Power Augmented Ram Engine in Cruise," *Chinese Journal of Aeronautics*, Vol. 23, No. 1, 2010, pp. 1-8. doi: Doi 10.1016/S1000-9361(09)60180-1
- [16] Qin, X. G., "Numerical Simulation of High Lift Device in Ground Effect," PhD thesis, Beijing University of Aeronautics and Astronautics, Beijing, 2010.
- [17] Deb, K., Pratap, A., Agarwal, S. and Meyarivan, T., "A Fast and Elitist Multi-objective Genetic Algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, Vol. 6, No. 2, 2002, pp. 182-197. doi: Pii S 1089-778x(02)04101-2. doi: 10.1109/4235.996017

- [18] Goldberg, D. E., *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, 1989.
- [19] Morgan, B., "Gairfoils: Finding High-Lift Joukowski Airfoils with a Genetic Algorithm," Technical Report, Dept. of Mechanical Engineering, Washington University in St. Louis, 2007.
- [20] Srinivas, N. and Deb, K., "Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms," *Evolutionary Computation*, Vol. 2, No. 3, 1994, pp. 221-248. doi: 10.1162/evco.1994.2.3.221
- [21] Konak, A., Coit, D. W. and Smith, A. E., "Multi-Objective Optimization Using Genetic Algorithms: A Tutorial," *Reliability Engineering & System Safety*, Vol. 91, No. 9, 2006, pp. 992-1007. doi: 10.1016/j.res.2005.11.018
- [22] Deb, K., "Single and Multi-Objective Optimization Using Evolutionary Computation," *Proceedings of the 6th International Conference on Hydro-informatics*. Singapore, 2004, pp. 14-35.
- [23] Durillo, J.J., Nebro, A.J. and Alba, E., "The jMetal Framework for Multi-Objective Optimization: Design and Architecture," *Proceedings of CEC 2010*, 2010, pp. 4138-4325.
- [24] Weiss, J. M. and Smith, W. A., "Preconditioning Applied to Variable and Constant Density Flows," *AIAA Journal*, Vol. 33, No. 11, 1995, pp. 2050-2057. doi: 10.2514/3.12946
- [25] Ramsay, R.R. and Gregorek, G. M., "Effects of Roughness and Pitch Oscillations on the S809 Airfoil," NREL Report, 1995.
- [26] Lanzafame, R. and Messina, M., "Fluid Dynamics of Wind Turbine Design: Critical Analysis, Optimization and Application of BEM Theory," *Renewable Energy*, Vol. 32, No. 14, 2007, pp. 2291-2305. doi: 10.1016/j.renene.2006.12.010
- [27] Glauert, H., *The Elements of Airfoil and Airscrew Theory*, Cambridge University Press, 1926.
- [28] Manwell, J. F. and McGowan, J. G., *Wind Energy Explained: Theory, Design and Application*, Wiley, 2010.

Appendix

Genetic Algorithm Code with Ancillary Files

The following is the code for the genetic algorithm based optimization method used in this thesis, as well as the script files of FLUENT and ICEM. The section names are the file names for the program.

A.1 NSGAI_main.java

```
package gaflatback;

public class NSGAI_main {
    public static Logger      logger_ ;      // Logger object
    public static FileHandler fileHandler_ ; // FileHandler object

    public static void main(String [] args) throws
        JMException,
        SecurityException,
        IOException,
        ClassNotFoundException {
        Problem   problem   ; // The problem to solve
        Algorithm algorithm ; // The algorithm to use
        Operator  crossover ; // Crossover operator
        Operator  mutation  ; // Mutation operator
        Operator  selection ; // Selection operator

        QualityIndicator indicators ; // Object to get quality indicators

        // Logger object and file to store log messages
        logger_      = Configuration.logger_ ;
        fileHandler_ = new FileHandler("NSGAI_main.log");
        logger_.addHandler(fileHandler_) ;

        indicators = null ;
        if (args.length == 1) { // did not understand
            Object [] params = {"Real"};
            problem = (new ProblemFactory()).getProblem(args[0],params);
        } // if
        else if (args.length == 2) {
            Object [] params = {"Real"};
            problem = (new ProblemFactory()).getProblem(args[0],params);
            indicators = new QualityIndicator(problem, args[1]) ;
        } // if

        else { // Default problem
            problem = new MOGA("Real", 24); // go to jmetal.problems for reference
            on how to contract a problem, and 12 means number of variables
        } // else
    }
}
```

```

algorithm = new NSGAI(problem);

// Algorithm parameters
algorithm.setInputParameter("populationSize",20); //maybe need to
change in the future
algorithm.setInputParameter("maxEvaluations",1000);

// Mutation and Crossover for Real codification
crossover = CrossoverFactory.getCrossoverOperator("SBXCrossover");
crossover.setParameter("probability",0.9);
crossover.setParameter("distributionIndex",20.0);

mutation = MutationFactory.getMutationOperator("PolynomialMutation");
mutation.setParameter("probability",1.0/problem.getNumberOfVariables());
mutation.setParameter("distributionIndex",20.0);

// Selection Operator
selection = SelectionFactory.getSelectionOperator("BinaryTournament2");

// Add the operators to the algorithm
algorithm.addOperator("crossover",crossover);
algorithm.addOperator("mutation",mutation);
algorithm.addOperator("selection",selection);

// Add the indicator object to the algorithm
algorithm.setInputParameter("indicators", indicators);

// Execute the Algorithm
long initTime = System.currentTimeMillis();
SolutionSet population = algorithm.execute();
    long estimatedTime = System.currentTimeMillis() - initTime;
logger_.info("Total execution time: "+estimatedTime + "ms");
logger_.info("Variables values have been written to file VAR");
population.printVariablesToFile("VAR");
logger_.info("Objectives values have been written to file FUN");
population.printObjectivesToFile("FUN");

if (indicators != null) {
    logger_.info("Quality indicators");
    logger_.info("Hypervolume: " + indicators.getHypervolume(population));
    logger_.info("GD : " + indicators.getGD(population));
    logger_.info("IGD : " + indicators.getIGD(population));
    logger_.info("Spread : " + indicators.getSpread(population));
    logger_.info("Epsilon : " + indicators.getEpsilon(population));

    int evaluations =
((Integer)algorithm.getOutputParameter("evaluations")).intValue();
    logger_.info("Speed : " + evaluations + " evaluations");
} // if
} //main
} // NSGAI_main

```

A.2 MOGA.java

```

package gaflatback;

public class MOGA extends Problem{

public MOGA(String solutionType, Integer numberOfVariables) throws
ClassNotFoundException{
    numberOfVariables_ = numberOfVariables.intValue() ;
    numberOfObjectives_ = 2 ;
    numberOfConstraints_ = 2 ;
    problemName_ = "NREL S809 Optimization" ;

    upperLimit_ = new double[numberOfVariables_] ;
    lowerLimit_ = new double[numberOfVariables_] ;

    // set the lower and upper bounds for x and y coordinates
    lowerLimit_[0] = 0.000; // x1
    upperLimit_[0] = 0.020;
    lowerLimit_[1] = 0.025; // x2
    upperLimit_[1] = 0.065;
    lowerLimit_[2] = 0.300; // x3
    upperLimit_[2] = 0.340;
    lowerLimit_[3] = 0.360; // x4
    upperLimit_[3] = 0.400;
    lowerLimit_[4] = 0.460; // x5
    upperLimit_[4] = 0.500;
    lowerLimit_[5] = 0.810; // x6
    upperLimit_[5] = 0.850;
    lowerLimit_[6] = 0.010; // y1
    upperLimit_[6] = 0.030;
    lowerLimit_[7] = 0.070; // y2
    upperLimit_[7] = 0.090;
    lowerLimit_[8] = 0.108; // y3
    upperLimit_[8] = 0.128;
    lowerLimit_[9] = 0.108; // y4
    upperLimit_[9] = 0.128;
    lowerLimit_[10] = 0.108; // y5
    upperLimit_[10] = 0.128;
    lowerLimit_[11] = 0.025; // y6
    upperLimit_[11] = 0.045;
    lowerLimit_[12] = 0.000; // m1
    upperLimit_[12] = 0.020;
    lowerLimit_[13] = 0.060; // m2
    upperLimit_[13] = 0.100;
    lowerLimit_[14] = 0.260; // m3
    upperLimit_[14] = 0.300;
    lowerLimit_[15] = 0.360; // m4
    upperLimit_[15] = 0.400;
    lowerLimit_[16] = 0.440; // m5
    upperLimit_[16] = 0.480;
    lowerLimit_[17] = 0.730; // m6
    upperLimit_[17] = 0.770;
    lowerLimit_[18] = -0.025; // n1
    upperLimit_[18] = -0.013;
    lowerLimit_[19] = -0.065; // n2
    upperLimit_[19] = -0.053;
    lowerLimit_[20] = -0.145; // n3
    upperLimit_[20] = -0.125;

```

```

        lowerLimit_[21] = -0.145; // n4
        upperLimit_[21] = -0.125;
        lowerLimit_[22] = -0.145; // n5
        upperLimit_[22] = -0.125;
        lowerLimit_[23] = 0.008; // n6
        upperLimit_[23] = 0.020;
        if (solutionType.compareTo("Real") == 0)
            solutionType_ = new RealSolutionType(this) ;
        else {
            System.out.println("Error: solution type " + solutionType + "
invalid") ;
            System.exit(-1) ;
        }
    }

    public void evaluate(Solution solution) throws JMException{ // the
algorithm itself is finding minimum
        double[] f =new double[numberOfObjectives_]; //f1 is 10/Cl, f2 is
100*Cd/Cl
        f[0]=1000;
        f[1]=1000;
        // compute f1 and f2 values
        double timeStep=0.02;
        ICEMAirfoilNRELS809 gt=new
ICEMAirfoilNRELS809(solution.getDecisionVariables());
        gt.buildAirfoil(timeStep);
        f=gt.getScoreWithFluent();
        solution.setObjective(0, f[0]);
        solution.setObjective(1, f[1]);
    }

    public void evaluateConstraints(Solution solution) throws JMException {
        double [] constraint = new double[this.getNumberOfConstraints()];
        double timeStep=0.02;
        ICEMAirfoilNRELS809 gt=new
ICEMAirfoilNRELS809(solution.getDecisionVariables());
        gt.buildAirfoil(timeStep);
        System.out.println(" max thickness = " + gt.getThickness()[1]);
        constraint[0] = gt.getThickness()[1] - 0.19;
        constraint[1] = 0.22 - gt.getThickness()[1];

        double total = 0.0;
        int number = 0;
        for (int i = 0; i < this.getNumberOfConstraints(); i++)
            if (constraint[i]<0.0){
                total+=constraint[i]; //total=total+constraint[i]
                number++; //number=number+1
            }

        solution.setOverallConstraintViolation(total);
        solution.setNumberOfViolatedConstraint(number);
    } // evaluateConstraints
}

```

A.3 AirfoilNRELS809.java

```
package gaflatback;

public class AirfoilNRELS809 {
    public double fitness;
    static int pointnumber = 6;
    public double[] X = new double[pointnumber];
    public double[] Y = new double[pointnumber];
    public double[] M = new double[pointnumber];
    public double[] N = new double[pointnumber];
    public static final double Xlead=0, Xtail=1;
    public static final double Ylead=0, Ytail=0;
    public static final double Mlead=0, Mtail=1;
    public static final double Nlead=0, Ntail=0;
    public static final double[] minX=new double[pointnumber];
    public static final double[] maxX=new double[pointnumber];
    public static final double[] minY=new double[pointnumber];
    public static final double[] maxY=new double[pointnumber];
    public static final double[] minM=new double[pointnumber];
    public static final double[] maxM=new double[pointnumber];
    public static final double[] minN=new double[pointnumber];
    public static final double[] maxN=new double[pointnumber];

    //Y points go from back to front and N points go from front to back
    public double[] getminX(){
        minX[0]=0;
        minX[1]=0.03;
        minX[2]=0.08;
        minX[3]=0.15;
        minX[4]=0.45;
        minX[5]=0.85;

        return minX;
    }

    public double[] getmaxX(){
        maxX[0]=0.03;
        maxX[1]=0.08;
        maxX[2]=0.15;
        maxX[3]=0.45;
        maxX[4]=0.85;
        maxX[5]=1;

        return maxX;
    }

    public double[] getminY(){
        minY[0]=0.01;
        minY[1]=0.06;
        minY[2]=0.10;
        minY[3]=0.10;
        minY[4]=0.10;
        minY[5]=0.015;
        return minY;
    }
}
```

```

}

    public double[] getmaxY(){
        maxY[0]=0.03;
        maxY[1]=0.09;
        maxY[2]=0.13;
        maxY[3]=0.13;
        maxY[4]=0.13;
        maxY[5]=0.05;
        return maxY;
}

    public double[] getminM(){
        minM[0]=0;
        minM[1]=0.03;
        minM[2]=0.10;
        minM[3]=0.21;
        minM[4]=0.35;
        minM[5]=0.70;
        return minM;
}

    public double[] getmaxM(){
        maxM[0]=0.03;
        maxM[1]=0.10;
        maxM[2]=0.21;
        maxM[3]=0.35;
        maxM[4]=0.70;
        maxM[5]=1.00;
        return maxM;
}

    public double[] getminN(){
        minN[0]=-0.02;
        minN[1]=-0.04;
        minN[2]=-0.10;
        minN[3]=-0.10;
        minN[4]=-0.10;
        minN[5]=0.015;
        return minN;
}

    public double[] getmaxN(){
        maxN[0]=-0.04;
        maxN[1]=-0.06;
        maxN[2]=-0.13;
        maxN[3]=-0.13;
        maxN[4]=-0.13;
        maxN[5]=0.032;
        return maxN;
}

/**Constructor for copying an existing airfoil**/
    public AirfoilNRELS809(AirfoilNRELS809 af){
        this(af.X,af.Y,af.M,af.N);
    }

/**Default constructor. Initialize everything to 0.**/
    public AirfoilNRELS809(){

```

```

    for(int i=0;i<pointnumber;i++){
        M[i]=0;
        N[i]=0;
    }
    for(int i=0;i<pointnumber;i++){
        X[i]=0;
        Y[i]=0;
    }

    fitness = 10000;    // maybe need to change in the future
}

/**Main constructor**/
public AirfoilNRELS809(double[] X, double[] Y, double[] M, double[] N){
    this.X=X;
    this.Y=Y;
    this.M=M;
    this.N=N;
}

public double[] getX(){
    return X;
}
public double[] getY(){
    return Y;
}
public double[] getM(){
    return M;
}
public double[] getN(){
    return N;
}
public double getFitness(){
    return fitness;
}
}

```


A.4 AirfoilNRELS809.java

```
public class ICEMAirfoilNRELS809 {
    static int pointnumber =6;
    public double fitness, thickness;
    public double[] xPoints, yPoints, mPoints, nPoints;
    public int iterations;
    public double max=0;           //?????????
    public static final double Xlead=0;
    public static final double Ylead=0;
    public static final double Xtail=1;
    public static final double Ytail=0;
    public static final double
Mlead=Xlead,Nlead=Ylead,Mtail=Xtail,Ntail=Ytail;
    public static final int extraPoints = 1;
    public double Ax,Bx,Cx,DDx,Ay,By,Cy,DDy,Dm,Em,Fm,Gm,Dn,En,Fn,Gn;
//??????????/
    public double[] X = new double[pointnumber];
    public double[] Y = new double[pointnumber];
    public double[] M = new double[pointnumber];
    public double[] N = new double[pointnumber];
    public int numberOfObjectives=2;

    public ICEMAirfoilNRELS809(AirfoilNRELS809 af){
        this.X = af.X;
        this.Y = af.Y;
        this.M = af.M;
        this.N = af.N;
    }

    /** constructor for MOGA */
    public ICEMAirfoilNRELS809(Variable[] af) throws JMException{
        this.X[0]=af[0].getValue();
        this.X[1]=af[1].getValue();
        this.X[2]=af[2].getValue();
        this.X[3]=af[3].getValue();
        this.X[4]=af[4].getValue();
        this.X[5]=af[5].getValue();
        this.Y[0]=af[6].getValue();
        this.Y[1]=af[7].getValue();
        this.Y[2]=af[8].getValue();
        this.Y[3]=af[9].getValue();
        this.Y[4]=af[10].getValue();
        this.Y[5]=af[11].getValue();
        this.M[0]=af[12].getValue();
        this.M[1]=af[13].getValue();
        this.M[2]=af[14].getValue();
        this.M[3]=af[15].getValue();
        this.M[4]=af[16].getValue();
        this.M[5]=af[17].getValue();
        this.N[0]=af[18].getValue();
        this.N[1]=af[19].getValue();
        this.N[2]=af[20].getValue();
        this.N[3]=af[21].getValue();
    }
}
```

```

        this.N[4]=af[22].getValue();
        this.N[5]=af[23].getValue();
    }

    public void buildAirfoil(double timeStep){
        iterations = (int) (1/0.01);
        xPoints = new double[iterations + extraPoints];
        yPoints = new double[iterations + extraPoints];
        mPoints = new double[iterations + extraPoints];
        nPoints = new double[iterations + extraPoints];
        double t = 0;
        //bezier curve first
        for(int i=0; i<3;i++){
            //System.out.println("y = " + Y[i]+" n = "+N[i]);
        }

        for(int i=0;i<iterations;i++){

            xPoints[i] = (Xlead*Math.pow(1-t,7)) +
(7*X[0]*t*Math.pow(1-t,6)) + (21*X[1]*Math.pow(t, 2)*Math.pow(1-t, 5)) +
(35*X[2]*Math.pow(t, 3)*Math.pow(1-t, 4))+(35*X[3]*Math.pow(t, 4)*Math.pow(1-
t, 3))+(21*X[4]*Math.pow(t, 5)*Math.pow(1-t, 2))+(7*X[5]*Math.pow(t,
6)*Math.pow(1-t, 1))+Xtail*Math.pow(t, 7);
            yPoints[i] = (Ylead*Math.pow(1-t,7)) +
(7*Y[0]*t*Math.pow(1-t,6)) + (21*Y[1]*Math.pow(t, 2)*Math.pow(1-t, 5)) +
(35*Y[2]*Math.pow(t, 3)*Math.pow(1-t, 4))+(35*Y[3]*Math.pow(t, 4)*Math.pow(1-
t, 3))+(21*Y[4]*Math.pow(t, 5)*Math.pow(1-t, 2))+(7*Y[5]*Math.pow(t,
6)*Math.pow(1-t, 1))+Ytail*Math.pow(t, 7);

            mPoints[i] = (Mlead*Math.pow(1-t,7)) +
(7*M[0]*t*Math.pow(1-t,6)) + (21*M[1]*Math.pow(t, 2)*Math.pow(1-t, 5)) +
(35*M[2]*Math.pow(t, 3)*Math.pow(1-t, 4))+(35*M[3]*Math.pow(t, 4)*Math.pow(1-
t, 3))+(21*M[4]*Math.pow(t, 5)*Math.pow(1-t, 2))+(7*M[5]*Math.pow(t,
6)*Math.pow(1-t, 1))+Mtail*Math.pow(t, 7);
            nPoints[i] = (Nlead*Math.pow(1-t,7)) +
(7*N[0]*t*Math.pow(1-t,6)) + (21*N[1]*Math.pow(t, 2)*Math.pow(1-t, 5)) +
(35*N[2]*Math.pow(t, 3)*Math.pow(1-t, 4))+(35*N[3]*Math.pow(t, 4)*Math.pow(1-
t, 3))+(21*N[4]*Math.pow(t, 5)*Math.pow(1-t, 2))+(7*N[5]*Math.pow(t,
6)*Math.pow(1-t, 1))+Ntail*Math.pow(t, 7);

            //System.out.println("yPoints = "+yPoints[i]+"
nPoints = "+nPoints[i]);
            /*xPoints[i] = (Ax*Math.pow(t,4)) +
(Bx*Math.pow(t,3)) + (Cx*Math.pow(t, 2)) + DDx*t+Xlead;
            yPoints[i] = (Ay*Math.pow(t,4)) + (By*Math.pow(t,3))
+ (Cy*Math.pow(t, 2)) + DDy*t+Ylead;
            mPoints[i] = (Dm*Math.pow(t,4)) + (Em*Math.pow(t,3))
+ (Fm*Math.pow(t, 2)) + Gm*t +Mlead;
            nPoints[i] = (Dn*Math.pow(t,4)) + (En*Math.pow(t,3))
+ (Fn*Math.pow(t, 2)) + Gn*t +Nlead;*/
            t = t + 0.01;
        }
        //Add tail points
        for(int j=iterations;j<iterations+extraPoints;j++){
            if(j==iterations){

```

```

        xPoints[iterations] = Xtail;
        yPoints[iterations] = Ytail;
        mPoints[iterations] = Mtail;
        nPoints[iterations] = Ntail;
    }
}

    public double[] getThickness(){//[1] is max thickness, [0] is min
thickness
    double max=0;
    double min=1000;
    double[] t=new double[xPoints.length] ;
    for (int i=0;i<xPoints.length;i++){
        t[i]=yPoints[i]-nPoints[i];
    }
    for(int i=0; i<t.length;i++){
        if(t[i]>max){
            max=t[i];
        }
        if(t[i]<min){
            min=t[i];
        }
    }
    double[] thickness=new double[2];
    thickness[0]=min;
    thickness[1]=max;
    // System.out.println("thickness = "+ thickness[1]);
    return thickness;
}

    public double getFitness(){
        return fitness;
    }

    public double[] getScoreWithFluent(/**int generation, int iteration,
double fitness**/){
        long maximumWait = 1000 * 60 * 40; // the last section
represent the actual waiting time in the unit of minutes
        double[] f=new double[numberOfObjectives];
        publishFile("NRELS809.dat"); // the code of publishFile is
in the below
        try{
            BufferedReader points = new BufferedReader(new
FileReader(new File("NRELS809.dat")));
            String pline;
            String check1;
            String check2;
            for(int i=0;i<88;i++){
                points.readLine();
            }
            pline = points.readLine();
            check1 = pline.substring(pline.indexOf(" ") + 1);
            check2 = check1.substring(0,check1.indexOf(" "));

```

```

        System.out.println("check is " + check2);
        if (check2=="NaN"){
            Process killmeshpoints =
Runtime.getRuntime().exec("F:\\Research\\Running\\GAWorkspace\\GAFX\\kills809
.bat");
            killmeshpoints.waitFor();
            System.out.println("ICEM DID NOT RETURN A
RESULT FOR THIS CASE BECAUSE OF WRONG VARIABLES");
        }else{
            Process ICEMProc =
Runtime.getRuntime().exec("NRELS809.bat"); //executive ICEM
            ICEMProc.waitFor();
        }
        boolean x = true;
        long time1 = System.currentTimeMillis();
        while(x){ // cleanup cl
and cd file after 6 seconds
            long interval1 = System.currentTimeMillis() -
time1;
            //System.out.println(" interval 1 = "+
interval1);
            File clhistory = new File("cl-history");
            File cdhistory = new File("cd-history");
            if(interval1 > 1000 * 60 * 0.1){
                Process cleanupProcl =
Runtime.getRuntime().exec("F:\\Research\\Running\\GAWorkspace\\GAFX\\cleanup.
.bat");
                cleanupProcl.waitFor();
                if((!clhistory.exists())&&(!cdhistory.exists())){
                    x=false;
                }
            }
        }
        if (check2=="NaN"){
            System.out.println("FLUENT CAN NOT BE LAUCHED
BECAUSE OF WRONG VARIABLES ");
        }else{
            Process fluentProc =
Runtime.getRuntime().exec("fluents809.bat");
            fluentProc.waitFor();
        }
        File transcript = new
File("F:\\Research\\Running\\GAWorkspace\\GAFX\\trans.jou");
        //wait and check for Fluent's return at one second
intervals
        long time = System.currentTimeMillis();
        while (true) {
            long interval = System.currentTimeMillis() - time;
            if( interval > maximumWait){
                publishFile("trans.jou");
                System.out.println("FLUENT DID NOT RETURN A RESULT
FOR THIS CASE ");
            }
        }
    }
}

```

```

        Process fluentKill =
Runtime.getRuntime().exec("F:\\Research\\Running\\GAWorkspace\\GAFX\\fluentKi
ll.bat");
        fluentKill.waitFor();
        return thousands();
    }
    if(transcript.exists()){
        BufferedReader cdInput = new
BufferedReader(new FileReader(new File("cd-history")));
        BufferedReader clInput = new
BufferedReader(new FileReader(new File("cl-history")));

        double cd = 0;
        double cl = 0;
        double cle=0;
        double cde=0;
        String linecl;
        String linedc;
        String clread;
        String cdread;
        String clneed;
        String cdneed;
        //skip the first two lines

        for(int i=0;i<51;i++){ //
5000 iteration
            clInput.readLine();
            cdInput.readLine();

            linecl = clInput.readLine();
            linedc = cdInput.readLine();
            //System.out.println(linecl);
            //System.out.println(linedc);
            clread =
linecl.substring(7,linecl.indexOf("e"));
            clneed =
linecl.substring(linecl.indexOf("e")+1);
            //System.out.println(clneed);
            cdread =
linedc.substring(7,linedc.indexOf("e"));
            cdneed =
linedc.substring(linedc.indexOf("e")+1);
            //System.out.println(cdneed);

            cl = Double.parseDouble(clread);
//convert string to value
            cle= Double.parseDouble(clneed);
            //System.out.println(cle);
            cl=cl*(Math.pow(10,cle));
            cd = Double.parseDouble(cdread);
            cde= Double.parseDouble(cdneed);
            //System.out.println(cde);
            cd=cd*(Math.pow(10,cde));
            if ((cl>0) && (cl<4) && (cd>0) && (cd<4)) {
                System.out.println("Cl = " + cl+"
Cd = "+cd);

```

```

        cl/cd);

        System.out.println("Cl/Cd = " +
            f[0]=10/cl;
            f[1]=100*cd/cl;

            return f;
        }else{
            System.out.println("Negative values,
return Cl/Cd = 100" );
            return thousands();
        }
    }
}
} catch(Exception e){
    e.printStackTrace();
    return thousands();
}

}

public double[] thousands(){
    double[] vec=new double[numberOfObjectives];
    vec[0]=1000;
    vec[1]=1000;
    return vec;
}

public boolean publishFile(String filename){
    try{
        // Create file
        FileWriter fstream = new FileWriter(filename);
        BufferedWriter out = new BufferedWriter(fstream);
        out.write((xPoints.length)+ " 2\r\n"); // \n is new
line
        for(int i=0;i<(xPoints.length);i++){
            out.write(xPoints[i]+ " " + yPoints[i] + " 0\r\n");
        }
        for(int i=0;i<(mPoints.length);i++){
            out.write(mPoints[i]+ " " + nPoints[i] + " 0\r\n");
        }
        out.close();
        return true;
    }
    catch (Exception e){
        System.err.println("Error: " + e.getMessage());
        return false;
    }
}
}
}

```

A.5 ReadRecord.java

```
public class ReadRecord {
    int numberOfSolutions;
    int numberOfObjectives;
    double X1[];
    double X2[];
    double X3[];
    double X4[];
    double X5[];
    double X6[];
    double Y1[];
    double Y2[];
    double Y3[];
    double Y4[];
    double Y5[];
    double Y6[];
    double M1[];
    double M2[];
    double M3[];
    double M4[];
    double M5[];
    double M6[];
    double N1[];
    double N2[];
    double N3[];
    double N4[];
    double N5[];
    double N6[];
    double Objective1[];
    double Objective2[];

    public ReadRecord(int numberOfSolutions){
        this.numberOfSolutions = numberOfSolutions;
        //this.numberOfObjectives = numberOfObjectives;
    }

    public void ReadVAR() {
        double X1[] = new double [numberOfSolutions];
        double X2[] = new double [numberOfSolutions];
        double X3[] = new double [numberOfSolutions];
        double X4[] = new double [numberOfSolutions];
        double X5[] = new double [numberOfSolutions];
        double X6[] = new double [numberOfSolutions];
        double Y1[] = new double [numberOfSolutions];
        double Y2[] = new double [numberOfSolutions];
        double Y3[] = new double [numberOfSolutions];
        double Y4[] = new double [numberOfSolutions];
        double Y5[] = new double [numberOfSolutions];
        double Y6[] = new double [numberOfSolutions];
        double M1[] = new double [numberOfSolutions];
        double M2[] = new double [numberOfSolutions];
        double M3[] = new double [numberOfSolutions];
    }
}
```

```

    double M4 []= new double [numberOfSolutions];
    double M5 []= new double [numberOfSolutions];
    double M6 []= new double [numberOfSolutions];
    double N1 []= new double [numberOfSolutions];
    double N2 []= new double [numberOfSolutions];
    double N3 []= new double [numberOfSolutions];
    double N4 []= new double [numberOfSolutions];
    double N5 []= new double [numberOfSolutions];
    double N6 []= new double [numberOfSolutions];
    try{
        BufferedReader InputRecord = new BufferedReader(new
FileReader(new File("VAR")));

        String wholeline []= new String [numberOfSolutions];

        /** read current generation */
        for (int j=0; j<numberOfSolutions; j++){
            wholeline[j]= InputRecord.readLine();
        }

        for (int k=0; k<numberOfSolutions; k++){ //make some
changes, but not understand X1[k] why is 10
            X1[k] =
Double.parseDouble(wholeline[k].substring(0,wholeline[k].indexOf(" ", 10)));
// indexOf means find the " " after the 10th element in the wholeline[k]
            System.out.println(" x1=" + X1[k]);
            X2[k] =
Double.parseDouble(wholeline[k].substring(wholeline[k].indexOf(" ",10)+1,
wholeline[k].indexOf(" ", 30)));
            System.out.println(" x2=" + X2[k]);
            X3[k] =
Double.parseDouble(wholeline[k].substring(wholeline[k].indexOf(" ",30)+1,
wholeline[k].indexOf(" ", 50)));
            System.out.println(" x3=" + X3[k]);
            X4[k] =
Double.parseDouble(wholeline[k].substring(wholeline[k].indexOf(" ",50)+1,
wholeline[k].indexOf(" ", 70)));
            System.out.println(" x4=" + X4[k]);
            X5[k] =
Double.parseDouble(wholeline[k].substring(wholeline[k].indexOf(" ",70)+1,
wholeline[k].indexOf(" ", 90)));
            System.out.println(" x5=" + X5[k]);
            X6[k] =
Double.parseDouble(wholeline[k].substring(wholeline[k].indexOf(" ",90)+1,
wholeline[k].indexOf(" ", 110)));
            System.out.println(" x6=" + X6[k]);
            Y1[k] =
Double.parseDouble(wholeline[k].substring(wholeline[k].indexOf(" ",110)+1,
wholeline[k].indexOf(" ", 130)));
            System.out.println(" y1=" + Y1[k]);
            Y2[k] =
Double.parseDouble(wholeline[k].substring(wholeline[k].indexOf(" ",130)+1,
wholeline[k].indexOf(" ", 150)));
            System.out.println(" y2=" + Y2[k]);
            Y3[k] =
Double.parseDouble(wholeline[k].substring(wholeline[k].indexOf(" ",150)+1,
wholeline[k].indexOf(" ", 170)));

```



```

        System.out.println(" y3=" + Y3[k]);
        Y4[k] =
Double.parseDouble(wholeline[k].substring(wholeline[k].indexOf(" ",170)+1,
wholeline[k].indexOf(" ", 190)));
        System.out.println(" y4=" + Y4[k]);
        Y5[k] =
Double.parseDouble(wholeline[k].substring(wholeline[k].indexOf(" ",190)+1,
wholeline[k].indexOf(" ", 210)));
        System.out.println(" y5=" + Y5[k]);
        Y6[k] =
Double.parseDouble(wholeline[k].substring(wholeline[k].indexOf(" ",210)+1,
wholeline[k].indexOf(" ", 230)));
        System.out.println(" y6=" + Y6[k]);
        M1[k] =
Double.parseDouble(wholeline[k].substring(wholeline[k].indexOf(" ",230)+1,
wholeline[k].indexOf(" ", 250)));
        System.out.println(" m1=" + M1[k]);
        M2[k] =
Double.parseDouble(wholeline[k].substring(wholeline[k].indexOf(" ",250)+1,
wholeline[k].indexOf(" ", 270)));
        System.out.println(" m2=" + M2[k]);
        M3[k] =
Double.parseDouble(wholeline[k].substring(wholeline[k].indexOf(" ",270)+1,
wholeline[k].indexOf(" ", 290)));
        System.out.println(" m3=" + M3[k]);
        M4[k] =
Double.parseDouble(wholeline[k].substring(wholeline[k].indexOf(" ",290)+1,
wholeline[k].indexOf(" ", 310)));
        System.out.println(" m4=" + M4[k]);
        M5[k] =
Double.parseDouble(wholeline[k].substring(wholeline[k].indexOf(" ",310)+1,
wholeline[k].indexOf(" ", 330)));
        System.out.println(" m5=" + M5[k]);
        M6[k] =
Double.parseDouble(wholeline[k].substring(wholeline[k].indexOf(" ",330)+1,
wholeline[k].indexOf(" ", 350)));
        System.out.println(" m6=" + M6[k]);
        N1[k] =
Double.parseDouble(wholeline[k].substring(wholeline[k].indexOf(" ",350)+1,
wholeline[k].indexOf(" ", 370)));
        System.out.println(" n1=" + N1[k]);
        N2[k] =
Double.parseDouble(wholeline[k].substring(wholeline[k].indexOf("
",370)+1,wholeline[k].indexOf(" ",390)));
        System.out.println(" n2=" + N2[k]);
        N3[k] =
Double.parseDouble(wholeline[k].substring(wholeline[k].indexOf(" ",390)+1,
wholeline[k].indexOf(" ",410)));
        System.out.println(" n3=" + N3[k]);
        N4[k] =
Double.parseDouble(wholeline[k].substring(wholeline[k].indexOf(" ",410)+1,
wholeline[k].indexOf(" ",430)));
        System.out.println(" n4=" + N4[k]);
        N5[k] =
Double.parseDouble(wholeline[k].substring(wholeline[k].indexOf(" ",430)+1,
wholeline[k].indexOf(" ",450)));
        System.out.println(" n5=" + N5[k]);

```

```

        N6[k] =
Double.parseDouble(wholeline[k].substring(wholeline[k].indexOf(" ",450)+1,
wholeline[k].indexOf(" ",470)));
        System.out.println(" n6=" + N6[k]);
    }

    getX1(X1);
    getY1(Y1);
    getX2(X2);
    getY2(Y2);
    getX3(X1);
    getY3(Y1);
    getX4(X2);
    getY4(Y2);
    getX5(X1);
    getY5(Y1);
    getX6(X2);
    getY6(Y2);
    getM1(X1);
    getN1(Y1);
    getM2(X2);
    getN2(Y2);
    getM3(X1);
    getN3(Y1);
    getM4(X2);
    getN4(Y2);
    getM5(X1);
    getN5(Y1);
    getM6(X2);
    getN6(Y2);

    }
    catch(Exception e){
        e.printStackTrace();
        System.exit(-1);
    }
}

public void ReadFUN() {
    double[] Objective1 = new double[numberOfSolutions];
    double[] Objective2 = new double[numberOfSolutions];

    try{
        BufferedReader InputRecord = new BufferedReader(new
FileReader(new File("FUN")));
        String wholeline[]= new String[numberOfSolutions];
        /** read current generation */
        for (int j=0; j<numberOfSolutions; j++){
            wholeline[j]= InputRecord.readLine();
        }

        for (int k=0; k<numberOfSolutions; k++){
            Objective1[k] = -
Double.parseDouble(wholeline[k].substring(0,wholeline[k].indexOf(" ", 10)));
            System.out.println(" objective1=" + Objective1[k]);

```

```

        Objective2[k] = -
Double.parseDouble(wholeline[k].substring(wholeline[k].indexOf(" ",5)+1,
wholeline[k].indexOf(" ", 20)));
        System.out.println(" objective2=" + Objective2[k]);
    }

    getObjective1(Objective1);
    getObjective2(Objective2);
} catch (Exception e) {
    e.printStackTrace();
    System.exit(-1);
}
}

public double[] getX1(double[] X1) {
    this.X1=X1;
    return X1;
}
public double[] getY1(double[] Y1) {
    this.Y1=Y1;
    return Y1;
}
public double[] getX2(double[] X2) {
    this.X2=X2;
    return X2;
}
public double[] getY2(double[] Y2) {
    this.Y2=Y2;
    return Y2;
}
public double[] getX3(double[] X3) {
    this.X3=X3;
    return X3;
}
public double[] getY3(double[] Y3) {
    this.Y3=Y3;
    return Y3;
}
public double[] getX4(double[] X4) {
    this.X4=X4;
    return X4;
}
public double[] getY4(double[] Y4) {
    this.Y4=Y4;
    return Y4;
}
public double[] getX5(double[] X5) {
    this.X5=X5;
    return X5;
}
public double[] getY5(double[] Y5) {
    this.Y5=Y5;
    return Y5;
}
public double[] getX6(double[] X6) {
    this.X6=X6;
    return X6;
}

```

```

}
public double[] getY6(double[] Y6) {
    this.Y6=Y6;
    return Y6;
}
public double[] getM1(double[] M1) {
    this.M1=M1;
    return M1;
}
public double[] getN1(double[] N1) {
    this.N1=N1;
    return N1;
}
public double[] getM2(double[] M2) {
    this.M2=M2;
    return M2;
}
public double[] getN2(double[] N2) {
    this.N2=N2;
    return N2;
}
public double[] getM3(double[] M3) {
    this.M3=M3;
    return M3;
}
public double[] getN3(double[] N3) {
    this.N3=N3;
    return N3;
}
public double[] getM4(double[] M4) {
    this.M4=M4;
    return M4;
}
public double[] getN4(double[] N4) {
    this.N4=N4;
    return N4;
}
public double[] getM5(double[] M5) {
    this.M5=M5;
    return M5;
}
public double[] getN5(double[] N5) {
    this.N5=N5;
    return N5;
}
public double[] getM6(double[] M6) {
    this.M6=M6;
    return M6;
}
public double[] getN6(double[] N6) {
    this.N6=N6;
    return N6;
}

public double[] getObjectivel(double[] Objectivel) {
    this.Objectivel=Objectivel;
    return Objectivel;
}

```

```
    }  
    public double[] getObjectives2(double[] Objective2){  
        this.Objective2=Objective2;  
        return Objective2;  
    }  
}
```

A.6 WriteRecord.java

```
public class writerecord {

    int generation;
    int genSize;
    int evaluations;
    int populationSize;
    int generationNumber;
    SolutionSet population;

    public writerecord(SolutionSet population, int generation){
        this.population = population;
        this.evaluations = evaluations;
        this.populationSize = populationSize;
        this.generationNumber = generation;
        //generationNumber = evaluations/populationSize-1;
    }

    public void recordEverything(){
        for (int i=0;i< population.getMaxSize();i++){
            try{
                BufferedWriter recordWriter = new BufferedWriter(new
FileWriter(new File("record.txt"), true));
                recordWriter.write("generationNumber " +
generationNumber + " (" + i + ") " +
population.get(i).getDecisionVariables()[0].getValue() + ","
+
population.get(i).getDecisionVariables()[1].getValue() + "," +
population.get(i).getDecisionVariables()[2].getValue() + ","
+
population.get(i).getDecisionVariables()[3].getValue() + "," +
population.get(i).getDecisionVariables()[4].getValue() + ","
+
population.get(i).getDecisionVariables()[5].getValue() + "," +
population.get(i).getDecisionVariables()[6].getValue() + ","
+
population.get(i).getDecisionVariables()[7].getValue() + "," +
population.get(i).getDecisionVariables()[8].getValue() + ","
+
population.get(i).getDecisionVariables()[9].getValue() + "," +
population.get(i).getDecisionVariables()[10].getValue()+ ","
+
population.get(i).getDecisionVariables()[11].getValue() + "," +
population.get(i).getDecisionVariables()[12].getValue()+ ","
+
population.get(i).getDecisionVariables()[13].getValue() + "," +
population.get(i).getDecisionVariables()[14].getValue()+ ","
+
population.get(i).getDecisionVariables()[15].getValue() + "," +
population.get(i).getDecisionVariables()[16].getValue()+ ","
+
population.get(i).getDecisionVariables()[17].getValue() + "," +
population.get(i).getDecisionVariables()[18].getValue()+ ",")
            }
```

```

+
population.get(i).getDecisionVariables()[19].getValue() + "," +
population.get(i).getDecisionVariables()[20].getValue()+ ","
+
population.get(i).getDecisionVariables()[21].getValue() + "," +
population.get(i).getDecisionVariables()[22].getValue()+ ","
+
population.get(i).getDecisionVariables()[23].getValue() + "," +"\n"+
Fitness 1 = " + population.get(i).getObjective(0)
+ " Fitness 2 = " +
population.get(i).getObjective(1) + "\r\n");
recordWriter.close();
}
catch(Exception e){
e.printStackTrace();
}
}
}
}

```

A.7 NRELS809.Jou

```
(cx-gui-do cx-activate-item "MenuBar*ReadSubMenu*Mesh...")
(cx-gui-do cx-set-text-entry "Select File*FilterText" "D:\Research\NREL S809
Optimiziation\Running\S809mesh\*")
(cx-gui-do cx-activate-item "Select File*Apply")
(cx-gui-do cx-set-text-entry "Select File*Text" "s809.msh")
(cx-gui-do cx-activate-item "Select File*OK")
(cx-gui-do cx-activate-item "MenuBar*MeshMenu*Scale...")
(cx-gui-do cx-set-toggle-button "Scale Mesh*Frame2(Scaling)*Table2(Scaling)*ToggleBox1*Specify
Scaling Factors" #f)
(cx-gui-do cx-activate-item "Scale Mesh*Frame2(Scaling)*Table2(Scaling)*ToggleBox1*Specify Scaling
Factors")
(cx-gui-do cx-set-real-entry-list "Scale Mesh*Frame2(Scaling)*Table2(Scaling)*Frame3(Scaling
Factors)*RealEntry1(X)" '( 0.5))
(cx-gui-do cx-set-real-entry-list "Scale Mesh*Frame2(Scaling)*Table2(Scaling)*Frame3(Scaling
Factors)*RealEntry2(Y)" '( 0.5))
(cx-gui-do cx-activate-item "Scale Mesh*Frame2(Scaling)*Table2(Scaling)*PushButton4(Scale)")
(cx-gui-do cx-activate-item "Scale Mesh*PanelButtons*PushButton1(Close)")
(cx-gui-do cx-activate-item "MenuBar*AdaptMenu*Region...")
(cx-gui-do cx-set-real-entry-list "Region Adaption*Frame3(Input Coordinates)*RealEntry5(X Max)" '( 11))
(cx-gui-do cx-set-real-entry-list "Region Adaption*Frame3(Input Coordinates)*RealEntry6(Y Max)" '( 11))
(cx-gui-do cx-set-real-entry-list "Region Adaption*Frame3(Input Coordinates)*RealEntry2(Y Min)" '( 3.5))
(cx-gui-do cx-activate-item "Region Adaption*PanelButtons*PushButton1(Mark)")
(cx-gui-do cx-activate-item "Region Adaption*PanelButtons*PushButton2(Cancel)")
(cx-gui-do cx-activate-item "MenuBar*SeparateSubMenu*Faces...")
(cx-gui-do cx-set-toggle-button "Separate Face Zones*Frame1(Options)*ToggleBox1(Options)*Face" #f)
(cx-gui-do cx-activate-item "Separate Face Zones*Frame1(Options)*ToggleBox1(Options)*Face")
(cx-gui-do cx-set-list-selections "Separate Face Zones*Frame4*List4(Zones)" '( 1))
(cx-gui-do cx-activate-item "Separate Face Zones*Frame4*List4(Zones)")
```



```

(cx-gui-do cx-set-toggle-button "Separate Face Zones*Frame1(Options)*ToggleBox1(Options)*Mark" #f)
(cx-gui-do cx-activate-item "Separate Face Zones*Frame1(Options)*ToggleBox1(Options)*Mark")
(cx-gui-do cx-set-list-selections "Separate Face Zones*Frame3*List3(Registers)" '( 0))
(cx-gui-do cx-activate-item "Separate Face Zones*Frame3*List3(Registers)")
(cx-gui-do cx-activate-item "Separate Face Zones*PanelButtons*PushButton1(OK)")
(cx-gui-do cx-activate-item "Separate Face Zones*PanelButtons*PushButton2(Cancel)")
(cx-gui-do cx-activate-item "MenuBar*MeshMenu*Rotate...")
(cx-gui-do cx-set-real-entry-list "Rotate Mesh*Frame1*Table1*RealEntry1(Rotation Angle)" '( -
0.059288))
(cx-gui-do cx-activate-item "Rotate Mesh*PanelButtons*PushButton1(OK)")
(cx-gui-do cx-activate-item "Rotate Mesh*PanelButtons*PushButton2(Cancel)")
(cx-gui-do cx-activate-item "MenuBar*DisplayMenu*Mesh...")
(cx-gui-do cx-activate-item "Mesh Display*PanelButtons*PushButton1(OK)")
(cx-gui-do cx-activate-item "Mesh Display*PanelButtons*PushButton2(Cancel)")
(cx-gui-do cx-activate-item "ToolBar*View Tools*autoscale")
(cx-gui-do cx-activate-item "NavigationPane*Frame1*PushButton5(Models)")
(cx-gui-do cx-set-list-selections "Models*Frame1*Table1*Frame1*List1(Models)" '( 2))
(cx-gui-do cx-activate-item "Models*Frame1*Table1*Frame1*List1(Models)")
(cx-gui-do cx-activate-item "Models*Frame1*Table1*PushButton2(Edit)")
(cx-gui-do cx-set-toggle-button "Viscous Model*Frame1*Table1*Frame1(Model)*ToggleBox1(Model)*k-
omega (2 eqn)" #f)
(cx-gui-do cx-activate-item "Viscous Model*Frame1*Table1*Frame1(Model)*ToggleBox1(Model)*k-
omega (2 eqn)")
(cx-gui-do cx-set-toggle-button "Viscous Model*Frame1*Table1*Frame7(k-omega
Model)*ToggleBox7(k-omega Model)*SST" #f)
(cx-gui-do cx-activate-item "Viscous Model*Frame1*Table1*Frame7(k-omega Model)*ToggleBox7(k-
omega Model)*SST")
(cx-gui-do cx-set-toggle-button "Viscous Model*Frame1*Table1*Frame9(k-omega
Options)*ToggleBox9(k-omega Options)*CheckButton2(Low-Re Corrections)" #f)
(cx-gui-do cx-activate-item "Viscous Model*Frame1*Table1*Frame9(k-omega Options)*ToggleBox9(k-
omega Options)*CheckButton2(Low-Re Corrections)")

```

```

(cx-gui-do cx-set-toggle-button "Viscous
Model*Frame1*Table1*Frame17(Options)*ToggleBox17(Options)*CheckBox4(Curvature
Correction)" #f)

(cx-gui-do cx-activate-item "Viscous
Model*Frame1*Table1*Frame17(Options)*ToggleBox17(Options)*CheckBox4(Curvature
Correction)")

(cx-gui-do cx-activate-item "Viscous Model*PanelButtons*PushButton1(OK)")

(cx-gui-do cx-activate-item "NavigationPane*Frame1*PushButton9(Boundary Conditions)")

(cx-gui-do cx-set-list-selections "Boundary Conditions*Frame1*Table1*Frame1*List1(Zone)" '( 1))

(cx-gui-do cx-activate-item "Boundary Conditions*Frame1*Table1*Frame1*List1(Zone)")

(cx-gui-do cx-set-list-selections "Boundary
Conditions*Frame1*Table1*Frame2*Table2*DropDownList2(Type)" '( 12))

(cx-gui-do cx-activate-item "Boundary
Conditions*Frame1*Table1*Frame2*Table2*DropDownList2(Type)")

(cx-gui-do cx-set-real-entry-list "velocity-inlet-14-
1*Frame4*Frame1(Momentum)*Frame1*Table1*Frame3(Turbulence)*Table3(Turbulence)*RealEntry9(
Turbulent Intensity)" '( 0.01))

(cx-gui-do cx-set-list-selections "velocity-inlet-14-
1*Frame4*Frame1(Momentum)*Frame1*Table1*DropDownList5(Velocity Specification Method)" '( 1))

(cx-gui-do cx-activate-item "velocity-inlet-14-
1*Frame4*Frame1(Momentum)*Frame1*Table1*DropDownList5(Velocity Specification Method)")

(cx-gui-do cx-set-real-entry-list "velocity-inlet-14-
1*Frame4*Frame1(Momentum)*Frame1*Table1*Frame11*Table11*RealEntry2(X-Velocity)" '(
75.3712))

(cx-gui-do cx-activate-item "velocity-inlet-14-1*PanelButtons*PushButton1(OK)")

(cx-gui-do cx-set-list-selections "Boundary Conditions*Frame1*Table1*Frame1*List1(Zone)" '( 2))

(cx-gui-do cx-activate-item "Boundary Conditions*Frame1*Table1*Frame1*List1(Zone)")

(cx-gui-do cx-set-list-selections "Boundary
Conditions*Frame1*Table1*Frame2*Table2*DropDownList2(Type)" '( 10))

(cx-gui-do cx-activate-item "Boundary
Conditions*Frame1*Table1*Frame2*Table2*DropDownList2(Type)")

(cx-gui-do cx-activate-item "Question*OK")

```

```

(cx-gui-do cx-set-real-entry-list "pressure-outlet-2-
1*Frame4*Frame1(Momentum)*Frame1*Table1*Frame3(Turbulence)*Table3(Turbulence)*RealEntry9(
Backflow Turbulent Intensity)" '( 0.01))

(cx-gui-do cx-activate-item "pressure-outlet-2-1*PanelButtons*PushButton1(OK)")

(cx-gui-do cx-set-list-selections "Boundary Conditions*Frame1*Table1*Frame1*List1(Zone)" '( 4))

(cx-gui-do cx-activate-item "Boundary Conditions*Frame1*Table1*Frame1*List1(Zone)")

(cx-gui-do cx-set-list-selections "Boundary
Conditions*Frame1*Table1*Frame2*Table2*DropDownList2(Type)" '( 10))

(cx-gui-do cx-activate-item "Boundary
Conditions*Frame1*Table1*Frame2*Table2*DropDownList2(Type)")

(cx-gui-do cx-activate-item "Question*OK")

(cx-gui-do cx-set-real-entry-list "pressure-outlet-15-
1*Frame4*Frame1(Momentum)*Frame1*Table1*Frame3(Turbulence)*Table3(Turbulence)*RealEntry9(
Backflow Turbulent Intensity)" '( 0.01))

(cx-gui-do cx-activate-item "pressure-outlet-15-1*PanelButtons*PushButton1(OK)")

(cx-gui-do cx-set-list-selections "Boundary Conditions*Frame1*Table1*Frame1*List1(Zone)" '( 0))

(cx-gui-do cx-activate-item "Boundary Conditions*Frame1*Table1*Frame1*List1(Zone)")

(cx-gui-do cx-activate-item "NavigationPane*Frame1*PushButton11(Dynamic Mesh)")

(cx-gui-do cx-activate-item "NavigationPane*Frame1*PushButton12(Reference Values)")

(cx-gui-do cx-set-list-selections "Reference Values*DropDownList1(Compute from)" '( 5))

(cx-gui-do cx-activate-item "Reference Values*DropDownList1(Compute from)")

(cx-gui-do cx-set-real-entry-list "Reference Values*Frame2(Reference Values)*Table2(Reference
Values)*RealEntry1(Area)" '( 0.5))

(cx-gui-do cx-activate-item "Reference Values*Frame2(Reference Values)*Table2(Reference
Values)*RealEntry1(Area)")

(cx-gui-do cx-set-real-entry-list "Reference Values*Frame2(Reference Values)*Table2(Reference
Values)*RealEntry5(Length)" '( 0.5))

(cx-gui-do cx-activate-item "Reference Values*Frame2(Reference Values)*Table2(Reference
Values)*RealEntry5(Length)")

(cx-gui-do cx-activate-item "NavigationPane*Frame1*PushButton9(Boundary Conditions)")

(cx-gui-do cx-set-list-selections "Boundary Conditions*Frame1*Table1*Frame1*List1(Zone)" '( 1))

(cx-gui-do cx-activate-item "Boundary Conditions*Frame1*Table1*Frame1*List1(Zone)")

```

```

(cx-gui-do cx-set-list-selections "Boundary
Conditions*Frame1*Table1*Frame2*Table2*DropDownList2(Type)" '( 12))

(cx-gui-do cx-activate-item "Boundary
Conditions*Frame1*Table1*Frame2*Table2*DropDownList2(Type)")

(cx-gui-do cx-set-real-entry-list "velocity-inlet-14-
1*Frame4*Frame1(Momentum)*Frame1*Table1*Frame11*Table11*RealEntry2(X-Velocity)" '(
75.3712))

(cx-gui-do cx-activate-item "velocity-inlet-14-1*PanelButtons*PushButton1(OK)")

(cx-gui-do cx-activate-item "NavigationPane*Frame1*PushButton12(Reference Values)")

(cx-gui-do cx-set-list-selections "Reference Values*DropDownList1(Compute from)" '( 5))

(cx-gui-do cx-activate-item "Reference Values*DropDownList1(Compute from)")

(cx-gui-do cx-activate-item "NavigationPane*Frame1*PushButton14(Solution Methods)")

(cx-gui-do cx-set-list-selections "Solution Methods*Frame1*Table1*Frame3(Spatial
Discretization)*Table3(Spatial Discretization)*DropDownList4(Turbulent Kinetic Energy)" '( 2))

(cx-gui-do cx-activate-item "Solution Methods*Frame1*Table1*Frame3(Spatial
Discretization)*Table3(Spatial Discretization)*DropDownList4(Turbulent Kinetic Energy)")

(cx-gui-do cx-set-list-selections "Solution Methods*Frame1*Table1*Frame3(Spatial
Discretization)*Table3(Spatial Discretization)*DropDownList5(Specific Dissipation Rate)" '( 1))

(cx-gui-do cx-activate-item "Solution Methods*Frame1*Table1*Frame3(Spatial
Discretization)*Table3(Spatial Discretization)*DropDownList5(Specific Dissipation Rate)")

(cx-gui-do cx-set-list-selections "Solution Methods*Frame1*Table1*Frame3(Spatial
Discretization)*Table3(Spatial Discretization)*DropDownList4(Turbulent Kinetic Energy)" '( 1))

(cx-gui-do cx-activate-item "Solution Methods*Frame1*Table1*Frame3(Spatial
Discretization)*Table3(Spatial Discretization)*DropDownList4(Turbulent Kinetic Energy)")

(cx-gui-do cx-activate-item "NavigationPane*Frame1*PushButton15(Solution Controls)")

(cx-gui-do cx-set-real-entry-list "Solution Controls*Frame1*Table1*Frame5(Under-Relaxation
Factors)*Table5(Under-Relaxation Factors)*RealEntry4(Momentum)" '( 0.35))

(cx-gui-do cx-activate-item "Solution Controls*Frame1*Table1*Frame5(Under-Relaxation
Factors)*Table5(Under-Relaxation Factors)*RealEntry4(Momentum)")

(cx-gui-do cx-activate-item "NavigationPane*Frame1*PushButton16(Monitors)")

(cx-gui-do cx-set-list-selections "Monitors*Frame1*Table1*Frame1*List1(Residuals, Statistic and Force
Monitors)" '( 0))

(cx-gui-do cx-activate-item "Monitors*Frame1*Table1*Frame1*List1(Residuals, Statistic and Force
Monitors)")

```

```

(cx-gui-do cx-activate-item "Monitors*Frame1*Table1*Frame2*Table2*PushButton2(Edit)")

(cx-gui-do cx-set-list-selections "Residual
Monitors*Frame1*Table1*Frame2*Table2*Frame2*Table2*Frame1*Table1*DropDownList3(Convergen
ce Criterion)" '( 1))

(cx-gui-do cx-activate-item "Residual
Monitors*Frame1*Table1*Frame2*Table2*Frame2*Table2*Frame1*Table1*DropDownList3(Convergen
ce Criterion)")

(cx-gui-do cx-activate-item "Residual Monitors*PanelButtons*PushButton1(OK)")

(cx-gui-do cx-activate-item "Monitors*Frame1*Table1*Frame2*Table2*PushButton1(Create)")

(cx-gui-do cx-activate-item "MenuBar*PopupMenuCreateMonitor*Drag...")

(cx-gui-do cx-set-toggle-button "Force Moment
Monitor*Frame1*Table1*Frame1*Table1*Frame1(Options)*Table1(Options)*CheckBox2(Plot)" #f)

(cx-gui-do cx-activate-item "Force Moment
Monitor*Frame1*Table1*Frame1*Table1*Frame1(Options)*Table1(Options)*CheckBox2(Plot)")

(cx-gui-do cx-set-toggle-button "Force Moment
Monitor*Frame1*Table1*Frame1*Table1*Frame1(Options)*Table1(Options)*CheckBox1(Print to
Console)" #f)

(cx-gui-do cx-activate-item "Force Moment
Monitor*Frame1*Table1*Frame1*Table1*Frame1(Options)*Table1(Options)*CheckBox1(Print to
Console)")

(cx-gui-do cx-set-list-selections "Force Moment
Monitor*Frame1*Table1*Frame2*Table2*Frame1*List1(Wall Zones)" '( 0))

(cx-gui-do cx-activate-item "Force Moment
Monitor*Frame1*Table1*Frame2*Table2*Frame1*List1(Wall Zones)")

(cx-gui-do cx-set-toggle-button "Force Moment
Monitor*Frame1*Table1*Frame1*Table1*Frame1(Options)*Table1(Options)*CheckBox4(Write)" #f)

(cx-gui-do cx-activate-item "Force Moment
Monitor*Frame1*Table1*Frame1*Table1*Frame1(Options)*Table1(Options)*CheckBox4(Write)")

(cx-gui-do cx-set-text-entry "Force Moment
Monitor*Frame1*Table1*Frame1*Table1*Frame1(Options)*Table1(Options)*Frame5*Table5*TextEntr
y2(File Name)" "D:\Research\NREL S809 Optimziation\Running\GAworkspace\GAFX\cd-history")

(cx-gui-do cx-activate-item "Force Moment Monitor*PanelButtons*PushButton1(OK)")

(cx-gui-do cx-activate-item "Monitors*Frame1*Table1*Frame2*Table2*PushButton1(Create)")

(cx-gui-do cx-activate-item "MenuBar*PopupMenuCreateMonitor*Lift...")

```

```

(cx-gui-do cx-set-toggle-button "Force Moment
Monitor*Frame1*Table1*Frame1*Table1*Frame1(Options)*Table1(Options)*CheckBox4(Write)" #f)

(cx-gui-do cx-activate-item "Force Moment
Monitor*Frame1*Table1*Frame1*Table1*Frame1(Options)*Table1(Options)*CheckBox4(Write)")

(cx-gui-do cx-set-text-entry "Force Moment
Monitor*Frame1*Table1*Frame1*Table1*Frame1(Options)*Table1(Options)*Frame5*Table5*TextEntry2(File Name)" "D:\Research\NREL S809 Optimziation\Running\GAworkspace\GAFX\cl-history")

(cx-gui-do cx-set-toggle-button "Force Moment
Monitor*Frame1*Table1*Frame1*Table1*Frame1(Options)*Table1(Options)*CheckBox2(Plot)" #f)

(cx-gui-do cx-activate-item "Force Moment
Monitor*Frame1*Table1*Frame1*Table1*Frame1(Options)*Table1(Options)*CheckBox2(Plot)")

(cx-gui-do cx-set-toggle-button "Force Moment
Monitor*Frame1*Table1*Frame1*Table1*Frame1(Options)*Table1(Options)*CheckBox1(Print to Console)" #f)

(cx-gui-do cx-activate-item "Force Moment
Monitor*Frame1*Table1*Frame1*Table1*Frame1(Options)*Table1(Options)*CheckBox1(Print to Console)")

(cx-gui-do cx-set-list-selections "Force Moment
Monitor*Frame1*Table1*Frame2*Table2*Frame1*List1(Wall Zones)" '( 0))

(cx-gui-do cx-activate-item "Force Moment
Monitor*Frame1*Table1*Frame2*Table2*Frame1*List1(Wall Zones)")

(cx-gui-do cx-activate-item "Force Moment Monitor*PanelButtons*PushButton1(OK)")

(cx-gui-do cx-activate-item "NavigationPane*Frame1*PushButton17(Solution Initialization)")

(cx-gui-do cx-set-toggle-button "Solution Initialization*Frame1*Table1*Frame3(Initialization Methods)*ToggleBox3(Initialization Methods)*Standard Initialization" #f)

(cx-gui-do cx-activate-item "Solution Initialization*Frame1*Table1*Frame3(Initialization Methods)*ToggleBox3(Initialization Methods)*Standard Initialization")

(cx-gui-do cx-set-list-selections "Solution Initialization*Frame1*Table1*DropDownList1(Compute from)" '( 4))

(cx-gui-do cx-activate-item "Solution Initialization*Frame1*Table1*DropDownList1(Compute from)")

(cx-gui-do cx-activate-item "Solution Initialization*Frame1*Table1*ButtonBox8*PushButton1(Initialize)")

(cx-gui-do cx-activate-item "NavigationPane*Frame1*PushButton19(Run Calculation)")

(cx-gui-do cx-set-integer-entry "Run Calculation*Frame1*Table1*IntegerEntry9(Number of Iterations)" 4500)

```

(cx-gui-do cx-activate-item "Run Calculation*Frame1*Table1*IntegerEntry9(Number of Iterations)")

(cx-gui-do cx-activate-item "Run Calculation*Frame1*Table1*PushButton21(Calculate)")

(cx-gui-do cx-activate-item "Information*OK")

(cx-gui-do cx-activate-item "MenuBar*WriteSubMenu*Stop Journal")

A.8 ICEMS809.rpl

ic_geo_cre_geom_input D:/Research/Running/GAworkspace/GAFX/Optimal(98_6.7).dat 0.001 input
PNTS pnt CRVS crv SURFS srf

ic_boco_solver

ic_boco_clear_icons

ic_csystem_display all 0

ic_csystem_set_current global

ic_boco_nastran_csystem reset

ic_geo_new_family GEOM

ic_boco_set_part_color GEOM

ic_point {} GEOM pnt.00 0,18,0

ic_point {} GEOM pnt.01 0,-18,0

ic_point {} GEOM pnt.02 -18,0,0

ic_point {} GEOM pnt.03 20,18,0

ic_point {} GEOM pnt.04 {20,-18\,0}

ic_point {} GEOM pnt.04 20,-18,0

ic_geo_new_family INLET

ic_boco_set_part_color INLET

ic_delete_geometry curve names crv.00 0

ic_curve point INLET crv.00 {pnt.03 pnt.00}

ic_delete_geometry curve names crv.01 0

ic_curve point INLET crv.01 {pnt.04 pnt.01}

ic_curve arc INLET crv.02 {pnt.00 pnt.02 pnt.01}

ic_geo_new_family OULET

ic_boco_set_part_color OULET

ic_delete_geometry curve names crv.03 0

ic_curve point OULET crv.03 {pnt.03 pnt.04}

ic_geo_new_family FLUID

ic_boco_set_part_color FLUID
ic_surface 2-4crvs FLUID srf.00 {0.01 {crv.02 crv.00 crv.01 crv.03}}
ic_set_dormant_pickable point 0 {}
ic_set_dormant_pickable curve 0 {}
ic_hex_initialize_mesh 2d new_numbering new_blocking FLUID
ic_hex_switch_blocking root
ic_hex_unblank_blocks
ic_hex_multi_grid_level 0
ic_hex_projection_limit 0
ic_hex_default_bunching_law default 2.0
ic_hex_floating_grid off
ic_hex_transfinite_degree 1
ic_hex_unstruct_face_type one_tri
ic_hex_set_unstruct_face_method uniform_quad
ic_hex_set_n_tetra_smoothing_steps 20
ic_hex_set_mesh_params PNTS CRVS SURFS GEOM INLET OULET FLUID -version 110
ic_hex_error_messages off_minor
ic_hex_switch_blocking root
ic_hex_set_edge_projection 19 21 0 1 crv.03
ic_hex_create_composite {crv.00 crv.02 crv.01}
ic_hex_set_edge_projection 13 21 0 1 crv.00
ic_hex_project_to_surface 13 21
ic_hex_set_edge_projection 11 13 0 1 crv.00
ic_hex_project_to_surface 11 13
ic_hex_set_edge_projection 11 19 0 1 crv.00
ic_hex_project_to_surface 11 19
ic_hex_mark_blocks unmark
ic_hex_mark_blocks superblock 4
ic_hex_mark_blocks numbers 19 21 edge_neighbors

ic_hex_ogrid 1 m PNTS CRVS SURFS GEOM INLET OULET FLUID -version 50
ic_hex_mark_blocks unmark
ic_hex_mark_blocks unmark
ic_hex_split_grid 33 35 pnt201 m PNTS CRVS SURFS GEOM INLET OULET FLUID VORFN
ic_hex_place_node 41 curve:crv.02 0.0515162585
ic_hex_place_node 41 curve:crv.00 0.989558101
ic_hex_place_node 39 curve:crv.01 0.989558101
ic_hex_mark_blocks unmark
ic_hex_mark_blocks superblock 4
ic_hex_change_element_id VORFN
ic_hex_set_edge_projection 33 42 0 1 crv0
ic_hex_project_to_surface 33 42
ic_hex_place_node 42 curve:crv0 1
ic_hex_place_node 33 curve:crv0 0.1
ic_hex_set_edge_projection 32 40 0 1 crv1
ic_hex_project_to_surface 32 40
ic_hex_place_node 40 curve:crv1 1
ic_hex_place_node 32 curve:crv1 0.1
ic_hex_mark_blocks unmark
ic_hex_mark_blocks superblock 17
ic_hex_collapse_blocks 1 version 410
ic_hex_create_composite {crv0 crv1}
ic_hex_set_edge_projection 32 33 0 1 crv0
ic_hex_project_to_surface 32 33
ic_hex_move_node 42 pnt201
ic_hex_place_node 41 curve:crv.00 0.94846
ic_hex_place_node 39 curve:crv.01 0.94846
ic_hex_place_node 13 curve:crv.02 0.25
ic_hex_place_node 11 curve:crv.02 0.75

ic_undo_group_begin
ic_hex_set_mesh 13 33 n 2 h1rel 0.0 h2rel 5.55015956709e-006 r1 2 r2 1.01 lmax 0 geo2
copy_to_parallel unlocked
ic_undo_group_end
ic_undo_group_begin
ic_hex_set_mesh 11 13 n 200 h1rel 0.0 h2rel 0.0 r1 2 r2 2 lmax 0 default copy_to_parallel unlocked
ic_undo_group_end
ic_undo_group_begin
ic_hex_set_mesh 11 13 n 200 h1rel 0.0 h2rel 0.0 r1 2 r2 2 lmax 0 uniform copy_to_parallel unlocked
ic_undo_group_end
ic_undo_group_begin
ic_hex_set_mesh 33 42 n 150 h1rel 0.0 h2rel 0.0 r1 2 r2 2 lmax 0 default copy_to_parallel unlocked
ic_undo_group_end
ic_undo_group_begin
ic_hex_set_mesh 33 42 n 150 h1rel 0.001086249279 h2rel 0.0002172498558 r1 2 r2 2 lmax 0 default
copy_to_parallel unlocked
ic_undo_group_end
ic_undo_group_begin
ic_hex_set_mesh 42 35 n 150 h1rel 0.0 h2rel 0.0 r1 2 r2 2 lmax 0 default copy_to_parallel unlocked
ic_undo_group_end
ic_undo_group_begin
ic_hex_set_mesh 42 35 n 150 h1rel 1.05263157895e-005 h2rel 0.0 r1 1.05 r2 2 lmax 0 default
copy_to_parallel unlocked
ic_undo_group_end
ic_hex_list_family_projection
ic_hex_create_mesh PNTS CRVS SURFS GEOM INLET OULET FLUID proj 2 dim_to_mesh 3
ic_undo_group_begin
ic_hex_set_mesh 13 33 n 150 h1rel 0.0 h2rel 5.55015956709e-006 r1 2 r2 1.01 lmax 0 geo2
copy_to_parallel unlocked
ic_undo_group_end

ic_undo_group_begin
ic_hex_set_mesh 13 33 n 150 h1rel 0.0 h2rel 5.55015956709e-006 r1 2 r2 1.01 lmax 0 geo2
copy_to_parallel unlocked
ic_undo_group_end
ic_hex_list_family_projection
ic_hex_create_mesh PNTS CRVS SURFS GEOM INLET OULET FLUID proj 2 dim_to_mesh 3
ic_undo_group_begin
ic_hex_split_grid 13 33 0.894632 m PNTS CRVS SURFS GEOM INLET OULET FLUID VORFN
ic_hex_list_family_projection
ic_hex_create_mesh PNTS CRVS SURFS GEOM INLET OULET FLUID proj 2 dim_to_mesh 3
ic_undo_group_end
ic_undo_group_begin
ic_hex_set_mesh 48 33 n 100 h1rel 0.0 h2rel 5.26739427012e-005 r1 2 r2 1.01 lmax 0 geo2
copy_to_parallel unlocked
ic_undo_group_end
ic_undo_group_begin
ic_hex_set_mesh 11 44 n 50 h1rel 0.0 h2rel 0.00930036457429 r1 2 r2 1.01 lmax 0 geo2
copy_to_parallel unlocked
ic_undo_group_end
ic_undo_group_begin
ic_hex_set_mesh 11 13 n 100 h1 0 h2 0 r1 2 r2 2 lmax 0 uniform unlocked
ic_undo_group_end
ic_undo_group_begin
ic_hex_set_mesh 44 45 n 80 h1 0.05 h2 0.0002 r1 2 r2 2 lmax 0 default unlocked
ic_undo_group_end
ic_undo_group_begin
ic_hex_set_mesh 32 42 n 80 h1 0.0022 h2 0.0002 r1 2 r2 2 lmax 0 default unlocked
ic_undo_group_end
ic_undo_group_begin
ic_hex_set_mesh 11 39 n 80 h1 0.4 h2 0.0002 r1 2 r2 2 lmax 0 default unlocked

ic_undo_group_end
ic_undo_group_begin
ic_hex_set_mesh 48 49 n 70 h1 0.03 h2 0.0002 r1 2 r2 2 lmax 0 default unlocked
ic_undo_group_end
ic_undo_group_begin
ic_hex_set_mesh 33 42 n 70 h1 0.002 h2 0.0002 r1 2 r2 2 lmax 0 default unlocked
ic_undo_group_end
ic_undo_group_begin
ic_hex_set_mesh 13 41 n 70 h1 0.35 h2 0.0002 r1 2 r2 2 lmax 0 default unlocked
ic_undo_group_end
ic_undo_group_begin
ic_hex_set_mesh 42 35 n 150 h1 0.0002 h2 0 r1 1.02 r2 2 lmax 0 geo1 unlocked
ic_undo_group_end
ic_undo_group_begin
ic_hex_set_mesh 49 50 n 80 h1 0.0002 h2 0 r1 1.02 r2 2 lmax 0 geo1 unlocked
ic_undo_group_end
ic_undo_group_begin
ic_hex_set_mesh 42 35 n 80 h1 0.0002 h2 0 r1 1.02 r2 2 lmax 0 geo1 unlocked
ic_undo_group_end
ic_undo_group_begin
ic_hex_set_mesh 45 46 n 80 h1 0.0002 h2 0 r1 2 r2 2 lmax 0 geo1 unlocked
ic_undo_group_end
ic_undo_group_begin
ic_hex_set_mesh 39 19 n 80 h1 0.0003 h2 0 r1 1.05 r2 2 lmax 0 geo1 unlocked
ic_undo_group_end
ic_undo_group_begin
ic_hex_set_mesh 41 21 n 80 h1 0.0003 h2 0 r1 1.05 r2 2 lmax 0 geo1 unlocked
ic_undo_group_end
ic_hex_list_family_projection

```

ic_hex_create_mesh PNTS CRVS SURFS GEOM INLET OULET FLUID proj 2 dim_to_mesh 2
ic_hex_write_file hex.uns PNTS CRVS SURFS GEOM INLET OULET FLUID proj 2 dim_to_mesh 2 -
family_boco family_boco.fbc
ic_uns_load hex.uns 3 0 {} 2
ic_uns_update_family_type visible {FLUID INLET GEOM OULET PNTS ORFN CRVS SURFS} {!NODE !LINE_2
QUAD_4} update 0
ic_uns_diag_reset_degen_min_max
ic_boco_solver
ic_uns_update_family_type visible {FLUID INLET GEOM OULET PNTS ORFN CRVS SURFS} {!NODE !LINE_2
QUAD_4} update 0
ic_boco_clear_icons
ic_csystem_display all 0
ic_csystem_set_current global
ic_boco_nastran_csystem reset
ic_undo_group_begin
ic_boco_solver {ANSYS Fluent}
ic_solver_mesh_info {ANSYS Fluent}
ic_undo_group_end
ic_boco_set GEOM {{ 1 {color} 16663866 }}
ic_boco_set INLET {{ 1 {color} 16663101 }}
ic_boco_set FLUID {{ 1 {color} 12109107 }}
ic_boco_set ORFN {}
ic_boco_set PNTS {}
ic_boco_set OULET {{ 1 {color} 3407186 }}
ic_boco_set CRVS {}
ic_boco_set SURFS {}
ic_geo_new_family VORFN 0
ic_uns_new_family VORFN 1
ic_boco_set VORFN {}
ic_boco_solver

```

```

ic_boco_save D:/Research/Running/S809mesh/nastran.fbc
ic_boco_save_atr D:/Research/Running/S809mesh/nastran.atr
ic_chdir D:/Research/Running/S809mesh
ic_geo_delete_family CREATED_FACES
ic_save_tetin project1.tin 0 0 {} {} 0 0 1
ic_rename project1.uns project1.uns.bak
ic_uns_check_duplicate_numbers
ic_save_unstruct project1.uns 1 {} {} {}
ic_uns_set_modified 1
ic_hex_save_blocking project1.blk
ic_boco_solver
ic_boco_save project1.fbc
ic_boco_save_atr project1.atr
ic_cart_is_loaded
ic_save_project_file D:/Research/Running/S809mesh/project1.prj {array\ set\ file_name\ \{{ catia_dir
.\} { parts_dir.\} { domain_loaded 1} { cart_file_loaded 0} { cart_file {} } { domain_saved
project1.uns} { archive {} } { med_replay {} } { topology_dir.\} { ugparts_dir.\} { icons
{{$env(ICEM_ACN)/lib/ai_env/icons} {$env(ICEM_ACN)/lib/va/EZCAD/icons}
{$env(ICEM_ACN)/lib/icons} {$env(ICEM_ACN)/lib/va/CABIN/icons}} { tetin project1.tin} {
family_boco project1.fbc} { iges_dir.\} { solver_params_loaded 0} { attributes_loaded 0} {
project_lock {} } { attributes project1.atr} { domain project1.uns} { domains_dir.\} { settings_loaded
0} { settings project1.prj} { blocking project1.blk} { hexa_replay {} } { transfer_dir.\} { mesh_dir.\} {
family_topo {} } { gemsparts_dir.\} { family_boco_loaded 0} { tetin_loaded 0} { project_dir.\} {
topo_mulcad_out {} } { solver_params hex.par} \} array\ set\ options\ \{{ expert 1} { remote_path
{} } { tree_disp_quad 2} { tree_disp_pyra 0} { evaluate_diagnostic 0} { histo_show_default 1} {
select_toggle_corners 0} { remove_all 0} { keep_existing_file_names 0} { record_journal 0} {
edit_wait 0} { face_mode all} { select_mode all} { med_save_emergency_tetin 1} { user_name
yileihe} { diag_which all} { uns_warn_if_display 500000} { bubble_delay 1000} { external_num 1} {
tree_disp_tri 2} { apply_all 0} { temporary_directory {} } { flood_select_angle 0} { home_after_load
1} { project_active 0} { histo_color_by_quality_default 1} { undo_logging 1} { tree_disp_hexa 0} {
histo_solid_default 1} { host_name CFDlabTeslaIV} { xhidden_full 1} { editor {} } { mouse_color
orange} { clear_undo 1} { remote_acn {} } { remote_sh csh} { tree_disp_penta 0} { n_processors
1} { remote_host {} } { save_to_new 0} { quality_info Quality} { tree_disp_node 0} {
med_save_emergency_mesh 1} { redtext_color red} { tree_disp_line 0} { select_edge_mode 0} {
use_dlremote 0} { max_mesh_map_size 1024} { show_tris 1} { remote_user {} } { auto_save_views
1} { max_cad_map_size 512} { display_origin 0} { uns_warn_user_if_display 1000000} { detail_info
0} { win_java_help 0} { show_factor 1} { boundary_mode all} { clean_up_tmp_files 1} {

```

```

med_save_emergency_blocking 1}{ max_binary_tetin 0}{ tree_disp_tetra 0}\} array\ set\
disp_options\ \{\{ uns_dualmesh 0}{ uns_warn_if_display 500000}{ uns_normals_colored 0}{
uns_icons 0}{ uns_locked_elements 0}{ uns_shrink_npos 0}{ uns_node_type None}{
uns_icons_normals_vol 0}{ uns_bcfield 0}{ backup Wire}{ uns_nodes 0}{ uns_only_edges 0}{
uns_surf_bounds 0}{ uns_wide_lines 0}{ uns_vol_bounds 0}{ uns_displ_orient Triad}{
uns_orientation 0}{ uns_directions 0}{ uns_thickness 0}{ uns_shell_diagnostic 0}{ uns_normals
0}{ uns_couplings 0}{ uns_periodicity 0}{ uns_single_surfaces 0}{ uns_midside_nodes 1}{
uns_shrink 100}{ uns_multiple_surfaces 0}{ uns_no_inner 0}{ uns_enums 0}{ uns_disp Wire}{
uns_bcfield_name {}{ uns_color_by_quality 0}{ uns_changes 0}{ uns_cut_delay_count 1000}\}
{set icon_size1 24}{set icon_size2 35}{set thickness_defined 0}{set solver_type 1}{set solver_setup 0}
array\ set\ prism_values\ \{\{ n_triangle_smoothing_steps 5}{ min_smoothing_steps 6}{
first_layer_smoothing_steps 1}{ new_volume {}{ height {}{ prism_height_limit {}{
interpolate_heights 0}{ n_tetra_smoothing_steps 10}{ do_checks {}{ delete_standalone 1}{
ortho_weight 0.50}{ max_aspect_ratio {}{ ratio_max {}{ total_height {}{ use_prism_v10 0}{
intermediate_write 1}{ delete_base_triangles {}{ ratio_multiplier {}{ refine_prism_boundary 1}{
max_size_ratio {}{ triangle_quality {}{ max_prism_angle 180}{ tetra_smooth_limit 0.3}{
max_jump_factor 5}{ use_existing_quad_layers 0}{ layers 3}{ fillet 0.10}{ into_orphan 0}{
init_dir_from_prev {}{ blayer_2d 0}{ do_not_allow_sticking {}{ top_family {}{ law exponential}
{ min_smoothing_val 0.1}{ auto_reduction 0}{ stop_columns 1}{ stair_step 1}{
smoothing_steps 12}{ side_family {}{ min_prism_quality 0.01}{ ratio 1.2}\} {set
aie_current_flavor {} array\ set\ vid_options\ \{\{ wb_NS_to_subset 0}{ auxiliary 0}{ show_name
0}{ inherit 1}{ default_part GEOM}{ new_srf_topo 1}{ DelPerFlag 0}{ show_item_name 0}{
composite_tolerance 1.0}{ wb_import_scale_geo 0}{ replace 0}{ same_pnt_tol 1e-4}{ tdv_axes
1}{ vid_mode 0}{ DelBlkPerFlag 0}\} {set savedTreeVisibility {geomNode 1 geom_subsetNode 2
geomPointNode 0 geomCurveNode 2 geomSurfNode 0 meshNode 1 mesh_subsetNode 2
meshPointNode 0 meshLineNode 0 meshShellNode 2 meshQuadNode 2 blockingNode 1
block_subsetNode 2 block_vertNode 0 block_edgeNode 2 block_faceNode 0 block_blockNode 0
block_meshNode 0 topoNode 2 topo-root 2 partNode 2 part-CRVS 2 part-FLUID 2 part-GEOM 2 part-
INLET 2 part-OULET 2 part-PNTS 2 part-SURFS 2 part-VORFN 0}\} {set last_view {rot {0 0 0 1} scale
{18.4138200543 18.4138200543 18.4138200543} center {1.4026 0.0 0.0} pos {0 0 0}} array\ set\
cut_info\ \{\{ active 0}{ whole 1}\} array\ set\ hex_option\ \{\{ default_bunching_ratio 2.0}{
floating_grid 0}{ n_tetra_smoothing_steps 20}{ trfDeg 1}{ wr_hexa7 0}{ smooth_ogrid 0}{
find_worst 1-3}{ hexa_verbose_mode 0}{ old_eparams 0}{ uns_face_mesh_method
uniform_quad}{ multigrid_level 0}{ uns_face_mesh_one_tri}{ check_blk 0}{ proj_limit 0}{
check_inv 0}{ project_bspline 0}{ hexa_update_mode 1}{ default_bunching_law BiGeometric}\}
array\ set\ saved_views\ \{\{ views {}}\} {ICEM CFD}

```

```

ic_exec {D:/ANSYS Inc/v145/icemcfd/win64_amd/icemcfd/output-interfaces/fluent6} -dom
D:/Research/Running/S809mesh/project1.uns -b project1.fbc -dim2d ./s809

```

```
ic_uns_num_couplings
```

```
ic_undo_group_begin
```

```
ic_uns_create_diagnostic_edgelist 1
```



```
ic_uns_diagnostic subset all diag_type uncovered fix_fam FIX_UNCOVERED diag_verb {Uncovered faces}
fams {} busy_off 1 quiet 1
```

```
ic_uns_create_diagnostic_edgelist 0
```

```
ic_undo_group_end
```

```
exit
```

Vita

Yilei He

- Degrees** M.S. Mechanical Engineering, August 2014
B.S. Mechanical Engineering, June 2008
- Birth of Place** Suzhou, Jiangsu, China
- Publications** Y.L. He, Ramesh K. Agarwal , “Optimization of NREL S809 Wind Turbine Airfoil using Multi-Objective Genetic Algorithm”, AIAA Applied Aerodynamic Conference, Atlanta, 16-20 June 2014.

August 2014