

Washington University in St. Louis
Washington University Open Scholarship

Engineering and Applied Science Theses &
Dissertations

McKelvey School of Engineering

Summer 8-2015

Shape Optimization of Busemann-Type Biplane Airfoil for Drag Reduction under Nonlifting and Lifting Conditions Using Genetic Algorithms

Yi Tian

Washington University in St Louis

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



Part of the [Aerospace Engineering Commons](#)

Recommended Citation

Tian, Yi, "Shape Optimization of Busemann-Type Biplane Airfoil for Drag Reduction under Nonlifting and Lifting Conditions Using Genetic Algorithms" (2015). *Engineering and Applied Science Theses & Dissertations*. 103.
https://openscholarship.wustl.edu/eng_etds/103

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

Swami Karunamoorthy

SHAPE OPTIMIZATION OF BUSEMANN-TYPE BIPLANE AIRFOIL FOR DRAG
REDUCTION UNDER NONLIFTING AND LIFTING CONDITIONS USING GENETIC
ALGORITHMS

by

Yi Tian

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

August 2015

Saint Louis, Missouri

Contents

List of Figures	iii
List of Tables.....	vii
Acknowledgments.....	viii
Dedication.....	ix
Abstract	x
1 Introduction.....	1
1.1 Motivation.....	1
1.2 Background.....	2
1.3 Scope of the Thesis.....	3
2 Flow Field Simulation of Busemann Biplane Airfoil.....	4
2.1 Wave Drag of Supersonic Thin Airfoil.....	4
2.2 Advantageous Effects of Busemann Biplane Airfoil.....	6
2.2.1 Wave Reduction Effect	7
2.2.2 Wave Cancellation Effect.....	8
2.3 Flow Field Simulation of the Standard Diamond-Shaped Airfoil and the Baseline Busemann Biplane Airfoil under Design and Off-design Conditions	10
2.3.1 Mesh Generation.....	11
2.3.2 Steady-State Flow Field Simulation at $M < 1.7$	13
2.3.3 Flow Field Simulation during Acceleration and Deceleration.....	18
2.3.3.1 Flow Field of Busemann Biplane Airfoil during Acceleration	19
2.3.3.2 Flow Field of Busemann Biplane Airfoil during Deceleration.....	23
3 Optimization of Busemann-Type Biplane Airfoil	25
3.1 Genetic-Algorithm-Based Optimization Procedure.....	25
3.1.1 Overview of Genetic Algorithm	26
3.2 Shape Optimization of Busemann Biplane Airfoil under Nonlifting Condition	27
3.2.1 Application of Single-Objective Genetic Algorithm (SOGA).....	27
3.2.2 Airfoil Parameterization	29
3.2.3 Optimization Results	33
3.3 Shape Optimization of Busemann Biplane Airfoil under Lifting Condition.....	57
3.3.1 Application of Multi-objective Genetic Algorithm (MOGA)	57
3.3.2 Airfoil Parameterization	58
3.3.3 Optimization Results	62
4 Conclusion.....	78

Appendix A	Mother Function for Optimization Code	79
Appendix B	GAparams.m	81
Appendix C	MOGAgeneral.m	84
Appendix D	FitnessFunction.m	86
Appendix E	GenerateIndividuals.m	88
Appendix F	DaughterCurveGenerator.m.....	92
Appendix G	Crossover.m.....	93
Appendix H	Mutator.m.....	94
References	95
Vita	97

List of Figures

Figure 2.1: Linear decomposition of contributions to lift and drag of an arbitrary thin airfoil at small angle of attack [19].....	6
Figure 2.2: Configuration of the Busemann biplane airfoil concept [17]	7
Figure 2.3: Wave reduction effect of the biplane airfoil [17].....	9
Figure 2.4: Wave cancellation effect [17].....	9
Figure 2.5: Wave formation and pressure changes at design and off-design condition for the Busemann biplane airfoil [17].....	10
Figure 2.6: Computational domain for the Busemann biplane airfoil configuration.....	11
Figure 2.7: H-mesh generated in the computational domain around the Busemann biplane airfoil	12
Figure 2.8: Zoomed-in view of H-mesh generated in computational domain around the Busemann biplane airfoil.....	12
Figure 2.9: Configurations of standard diamond-shaped airfoil (left) and baseline Busemann biplane airfoil (right)	13
Figure 2.10: c_d comparison of two different airfoils under non-lifting condition.....	14
Figure 2.11: Velocity contours around the Busemann airfoil at $M = 0.8$	16
Figure 2.12: Velocity contours around the Busemann airfoil at $M = 1.6$	17
Figure 2.13: Pressure contours around the Busemann airfoil at $M = 0.8$	17
Figure 2.14: Pressure contours around the Busemann airfoil at $M = 1.6$	18
Figure 2.15: c_d plots for diamond-shaped airfoil and Busemann airfoil during acceleration and deceleration under non-lifting condition.....	19
Figure 2.16: c_p -contours around the Busemann biplane airfoil with zero-lift during acceleration	22
Figure 2.17: c_p -contours around the Busemann biplane airfoil with zero-lift during deceleration.....	24
Figure 3.1: Schematic of information flow in the optimization process	25
Figure 3.2: Randomly generated Busemann-type biplane airfoil shape using Bezier curves	31
Figure 3.3: Detailed geometry of Bezier curve #1 and its control points	32
Figure 3.4: Detailed geometry of Bezier curve #2 and its control points	32
Figure 3.5: Geometry of both the original and optimized Busemann biplane airfoil.....	33
Figure 3.6: Bezier curve #1 and its control points for the optimized Busemann biplane airfoil.....	34
Figure 3.7: Bezier curve #2 and its control points for the optimized Busemann biplane airfoil.....	34
Figure 3.8: c_p contours around the original Busemann biplane airfoil with zero-lift at $M = 1.1$ during acceleration.....	36
Figure 3.9: c_p contours around the original Busemann biplane airfoil with zero-lift at $M = 1.2$ during acceleration.....	36
Figure 3.10: c_p contours around the original Busemann biplane airfoil with zero-lift at $M = 1.3$ during acceleration.....	37
Figure 3.11: c_p contours around the original Busemann biplane airfoil with zero-lift at $M = 1.4$ during acceleration	37
Figure 3.12: c_p contours around the original Busemann biplane airfoil with zero-lift at $M = 1.5$ during acceleration	38
Figure 3.13: c_p contours around the original Busemann biplane airfoil	

with zero-lift at $M = 1.6$ during acceleration	38
Figure 3.14: c_p contours around the original Busemann biplane airfoil	
with zero-lift at $M = 1.7$ during acceleration	39
Figure 3.15: c_p contours around the original Busemann biplane airfoil	
with zero-lift at $M = 1.7$ during deceleration.....	39
Figure 3.16: c_p contours around the original Busemann biplane airfoil	
with zero-lift at $M = 1.6$ during deceleration.....	40
Figure 3.17: c_p contours around the original Busemann biplane airfoil	
with zero-lift at $M = 1.5$ during deceleration.....	40
Figure 3.18: c_p contours around the original Busemann biplane airfoil	
with zero-lift at $M = 1.4$ during deceleration.....	41
Figure 3.19: c_p contours around the original Busemann biplane airfoil	
with zero-lift at $M = 1.3$ during deceleration.....	41
Figure 3.20: c_p contours around the original Busemann biplane airfoil	
with zero-lift at $M = 1.2$ during deceleration.....	42
Figure 3.21: c_p contours around the original Busemann biplane airfoil	
with zero-lift at $M = 1.1$ during deceleration.....	42
Figure 3.22: c_p contours around the optimized Busemann biplane airfoil	
with zero-lift at $M = 1.1$ during acceleration	43
Figure 3.23: c_p contours around the optimized Busemann biplane airfoil	
with zero-lift at $M = 1.2$ during acceleration	43
Figure 3.24: c_p contours around the optimized Busemann biplane airfoil	
with zero-lift at $M = 1.3$ during acceleration	44
Figure 3.25: c_p contours around the optimized Busemann biplane airfoil	
with zero-lift at $M = 1.4$ during acceleration	44
Figure 3.26: c_p contours around the optimized Busemann biplane airfoil	
with zero-lift at $M = 1.5$ during acceleration	45
Figure 3.27: c_p contours around the optimized Busemann biplane airfoil	
with zero-lift at $M = 1.6$ during acceleration	45
Figure 3.28: c_p contours around the optimized Busemann biplane airfoil	
with zero-lift at $M = 1.7$ during acceleration	46
Figure 3.29: c_p contours around the optimized Busemann biplane airfoil	
with zero-lift at $M = 1.7$ during deceleration.....	46
Figure 3.30: c_p contours around the optimized Busemann biplane airfoil	
with zero-lift at $M = 1.6$ during deceleration.....	47
Figure 3.31: c_p contours around the optimized Busemann biplane airfoil	
with zero-lift at $M = 1.5$ during deceleration.....	47
Figure 3.32: c_p contours around the optimized Busemann biplane airfoil	
with zero-lift at $M = 1.4$ during deceleration.....	48
Figure 3.33: c_p contours around the optimized Busemann biplane airfoil	
with zero-lift at $M = 1.3$ during deceleration.....	48
Figure 3.34: c_p contours around the optimized Busemann biplane airfoil	

with zero-lift at $M = 1.2$ during deceleration.....	49
Figure 3.35: c_p contours around the optimized Busemann biplane airfoil with zero-lift at $M = 1.1$ during deceleration.....	49
Figure 3.36: c_d plot for different airfoils under nonlifting condition	51
Figure 3.37: c_p -contours of the GA-optimized Busemann biplane airfoil with zero-lift during acceleration	52
Figure 3.38: c_p -contours of the GA-optimized Busemann biplane airfoil with zero-lift during deceleration	53
Figure 3.39: c_p -contours of the Adjoint-based-optimized Busemann biplane airfoil with zero-lift during acceleration [17]	54
Figure 3.40: c_p -contours of the Adjoint-based-optimized Busemann biplane airfoil with zero-lift during deceleration [17]	55
Figure 3.41: c_d plot for different airfoils under nonlifting condition.....	56
Figure 3.42: Randomly generated Busemann-type biplane airfoil shape using Bezier curves for lifting condition	60
Figure 3.43: Detailed geometry of Bezier curve #1 and its control points	60
Figure 3.44: Detailed geometry of Bezier curve #2 and its control points	61
Figure 3.45: Detailed geometry of Bezier curve #3 and its control points	61
Figure 3.46: Detailed geometry of Bezier curve #4 and its control points	62
Figure 3.47: Geometry of both the original and optimized Busemann biplane airfoil under lifting condition.....	63
Figure 3.48: Bezier curve #1 and its control points for the optimized Busemann biplane airfoil under lifting condition.....	63
Figure 3.49: Bezier curve #2 and its control points for the optimized Busemann biplane airfoil under lifting condition.....	64
Figure 3.50: Bezier curve #3 and its control points for the optimized Busemann biplane airfoil under lifting condition.....	64
Figure 3.51: Bezier curve #4 and its control points for the optimized Busemann biplane airfoil under lifting condition.....	65
Figure 3.52: c_p contours around the optimized Busemann biplane airfoil under lifting condition at $M = 1.1$ during acceleration	66
Figure 3.53: c_p contours around the optimized Busemann biplane airfoil under lifting condition at $M = 1.2$ during acceleration	67
Figure 3.54: c_p contours around the optimized Busemann biplane airfoil under lifting condition at $M = 1.3$ during acceleration	67
Figure 3.55: c_p contours around the optimized Busemann biplane airfoil under lifting condition at $M = 1.4$ during acceleration	68
Figure 3.56: c_p contours around the optimized Busemann biplane airfoil under lifting condition at $M = 1.5$ during acceleration	68
Figure 3.57: c_p contours around the optimized Busemann biplane airfoil under lifting condition at $M = 1.6$ during acceleration	69
Figure 3.58: c_p contours around the optimized Busemann biplane airfoil under lifting condition at $M = 1.7$ during acceleration	69
Figure 3.59: c_p contours around the optimized Busemann biplane airfoil	

	under lifting condition at $M = 1.7$ during deceleration.....	70
Figure 3.60:	c_p contours around the optimized Busemann biplane airfoil	
	under lifting condition at $M = 1.6$ during deceleration.....	70
Figure 3.61:	c_p contours around the optimized Busemann biplane airfoil	
	under lifting condition at $M = 1.5$ during deceleration.....	71
Figure 3.62:	c_p contours around the optimized Busemann biplane airfoil	
	under lifting condition at $M = 1.4$ during deceleration.....	71
Figure 3.63:	c_p contours around the optimized Busemann biplane airfoil	
	under lifting condition at $M = 1.3$ during deceleration.....	72
Figure 3.64:	c_p contours around the optimized Busemann biplane airfoil	
	under lifting condition at $M = 1.2$ during deceleration.....	72
Figure 3.65:	c_p contours around the optimized Busemann biplane airfoil	
	under lifting condition at $M = 1.1$ during deceleration.....	73
Figure 3.66:	c_d plot for different airfoils under lifting condition	74
Figure 3.67:	c_p -contours of the GA-optimized Busemann biplane airfoil	
	with lift during acceleration.....	75
Figure 3.68:	c_p -contours of the GA-optimized Busemann biplane airfoil	
	with lift during deceleration	76

List of Tables

Table 2.1: Drag coefficient c_d for diamond-shaped airfoil and Busemann airfoils over a range of Mach numbers	15
Table 2.2: Drag Coefficient c_d for Busemann airfoil and standard diamond airfoil at various Mach numbers.....	21
Table 3.1: c_d comparison at zero lift condition during acceleration (1 count = 0.0001) [17]	28
Table 3.2: c_d comparison at zero lift condition during deceleration (1 count = 0.0001) [17]	28
Table 3.3: GA parameters for shape optimization of Busemann biplane airfoil under nonlifting condition.....	29
Table 3.4: Coordinate range for each control point of the Bezier curves for the airfoil under nonlifting condition	31
Table 3.5: c_d comparison for the original and optimized Busemann biplane airfoil at nonlifting condition during acceleration (1 count = 0.0001).....	35
Table 3.6: c_d comparison for the original and optimized Busemann biplane airfoil at nonlifting condition during deceleration (1 count = 0.0001)	35
Table 3.7: GA parameters for shape optimization of Busemann biplane airfoil under lifting condition	57
Table 3.8: Coordinate range for each control point of the Bezier curves for the upper component of the airfoil under lifting condition.....	58
Table 3.9: Coordinate range for each control point of the Bezier curves for the lower component of the airfoil under lifting condition	59
Table 3.10: c_d comparison for the original and optimized Busemann biplane airfoil at lifting condition during acceleration (1 count = 0.0001).....	66
Table 3.11: c_d comparison for the original and optimized Busemann biplane airfoil at lifting condition during deceleration (1 count = 0.0001)	66
Table 3.12: c_l for the optimized Busemann biplane airfoil under lifting condition at different design points	77

Acknowledgments

I would like to acknowledge Dr. Ramesh Agarwal for encouraging me to engage in my graduate research and guiding me through this whole process. His knowledge in computational fluid dynamics has greatly helped me completing this research. His attitude towards work and life has also inspired me to get better prepared for my future career.

I would also like to thank Chris Seager for his generous help on the GA optimization process and everyone in the CFD lab who has helped me throughout my research.

Yi Tian

Washington University in St. Louis

August 2015

Dedicated to my parents

I would like to dedicate this thesis to my beloved parents (Haiying Yu and Zuowei Tian) for their continuous unconditional support of my studies in the United States.

I love you forever!

ABSTRACT

SHAPE OPTIMIZATION OF BUSEMANN-TYPE BIPLANE AIRFOIL FOR DRAG REDUCTION UNDER NONLIFTING AND LIFTING CONDITIONS USING GENETIC ALGORITHMS

by

Yi Tian

Master of Science in Mechanical Engineering

Washington University in St. Louis, 2015

Research Advisor: Professor Ramesh K. Agarwal

The focus of this thesis is on the shape optimization of the Busemann-type biplane airfoil for drag reduction under both nonlifting and lifting conditions using genetic algorithms. The concept of the Busemann-type biplane airfoil was first introduced by Adolf Busemann in 1935. Under its design condition at a specific supersonic flow speed, the Busemann biplane airfoil eliminates all wave drag due to its symmetrical biplane configuration; however it produces zero lift. Previous research has shown that the original Busemann biplane airfoil design has a poor performance under off-design conditions as well. In order to solve this problem of zero lift and to improve the off-design-condition performance of the biplane airfoil, shape optimization of the asymmetric biplane airfoil is performed to minimize the drag while maximizing the lift. In this thesis, the commercially available CFD solver ANSYS FLUENT is employed for computing the inviscid flow past the biplane airfoil. An unstructured mesh is generated using ICEM software. A second-order accurate steady density-based solver is employed to compute the supersonic flow field. A single-objective genetic algorithm (SOGA) is employed to optimize the Busemann biplane airfoil shape under nonlifting condition to

minimize the drag coefficient and a multi-objective genetic algorithm (MOGA) is employed to optimize the Busemann biplane airfoil shape under lifting condition to maximize both the lift coefficient and the lift to drag ratio simultaneously. Both results obtained by using SOGA and MOGA show significant improvement in the design and off-design-condition performance of the optimized Busemann biplane airfoil compared to the original one.

Chapter 1

Introduction

This chapter provides the background of the supersonic airfoils of commercial airplanes and the motivation behind the study of the optimization of the shape of such airfoils. The scope of this thesis is also included.

1.1 Motivation

For decades, the speed of commercial aircrafts has been bounded by the sound barrier. Even the most successful supersonic transport (SST) plane, the Concorde, could only be deployed in very few routes due to government regulation, low efficiency and excessive noise generation. Since the retirement of Concorde in 2003, the desire of developing a replacement for Concorde still remains. In order to accomplish that, one of the biggest design challenges is to eliminate, or at least greatly reduce, the strong bow shock wave generated during supersonic flight, which can cause high wave drag and substantial noise. This research employs a Single-Objective Genetic Algorithm (SOGA) and a Multi-Objective Genetic Algorithm (MOGA) to optimize the airfoil shape to minimize the wave drag of supersonic commercial aircrafts.

1.2 Background

Computational Fluid Dynamics (CFD) is routinely used in a variety of industrial applications as an analysis tool to simulate and validate new concepts and designs of systems and components that involve fluid flow including both liquid and gas. Components such as wings, fuselage, tails and engines are routinely analyzed for both commercial and fighter aircrafts using CFD.

At supersonic speed, a bow shock is generated ahead of the airplane. This shock wave generates a substantially high wave drag which needs to be overcome by the engine by providing a much higher thrust compared to that for a conventional subsonic/transonic airplane, which results in higher fuel consumption and low propulsive efficiency. In order to address the high-wave-drag problem at supersonic speed, a biplane concept was proposed by Adolf Busemann in 1935 [1] which can potentially avoid the formation of the bow shock and thus does not create sonic boom. During the period from 1935 to 1960, substantial research was conducted on the Busemann biplane concept. Moeckel [2] and Licher [3] performed the theoretical analysis of the optimized lifting supersonic biplanes. Tan [4] took a further step and derived analytical expressions for the drag and lift of a three dimensional supersonic biplane with a finite span of rectangular plan form. Some experimental results were obtained by Ferri [5] using a wind tunnel and comparisons were made between the experimental and analytical results. During the past ten years, considerable interest has again generated in supersonic biplane airfoils. Igra and Arad [6] tested and analyzed the effects of different parameters on the drag coefficient of the Busemann airfoil at various flow conditions. Kusunose et al. [7] proposed a concept for the next generation supersonic transport using the Busemann biplane design. A series of studies using both computational fluid dynamics (CFD) methods and wind-tunnel experimental methods have been completed by Kusunose's research group [7-16]. Recently,

Hu et al. [17] employed a multi-point adjoint-based aerodynamic design and optimization method to improve the baseline Busemann biplane airfoil's off-design performance and alleviate the flow hysteresis problem. They also addressed the problem of minimizing its drag by shape optimization.

1.3 Scope of the Thesis

This thesis addresses the aerodynamic shape optimization of the two-dimensional symmetric Busemann biplane airfoil under both design and off-design conditions for reducing its wave drag and the alleviation of the flow-hysteresis and choked-flow effects using a single-objective genetic algorithm. The symmetric Busemann biplane airfoil generates zero lift; this problem is addressed by introducing asymmetry in the shape of the two profiles of the biplane. The asymmetric configuration is shape optimized for maximum lift and minimum drag by employing a multi-objective genetic algorithm. The flow field is computed using the commercial CFD software ANSYS FLUENT. Body-fitted H-grids around the airfoils are generated using the ICEM software. Random airfoil shapes with constraints in a given generation of the genetic algorithm are generated by employing the Bezier Curves (third-order polynomials).

Chapter 2

Flow Field Simulation of Busemann Biplane Airfoil

This chapter presents the physical characteristics of the Busemann-type biplane airfoil and the simulation of supersonic flow past the Busemann biplane airfoil under both design and off-design conditions. A total of three types of flows are simulated: an impulsive uniform flow; a flow during acceleration; and a flow during deceleration.

2.1 Wave Drag of Supersonic Thin Airfoil

When measuring the total drag of an airplane wing, several types of drag are taken into consideration: skin friction drag, wave drag, pressure drag and vortex drag. However in supersonic cruise flight, the wave drag, which is the drag caused by the existence of strong shock waves, is the dominant component among all others [17].

In order to solve for the lift and drag of an airfoil in supersonic flow, the shock expansion theory is generally employed. Under the conditions that the airfoil is thin and the angle of attack is small, the lift and drag of an airfoil in supersonic flow can be approximately determined by the analytical expressions using the thin airfoil theory [19]. The lift and drag coefficients of an airfoil are defined in Eq. (2.1) and Eq. (2.2).

$$c_l = \frac{L}{qc} \quad (2.1)$$

$$c_d = \frac{D}{qc} \quad (2.2)$$

where L is the lift, D is the wave drag of the airfoil, c is the chord length and q is the dynamic pressure defined as

$$q = \frac{1}{2} \rho_\infty U_\infty^2 \quad (2.3)$$

It should be noted that in Eq. (2.1) and Eq. (2.2), a wingspan of unit length is assumed. According to thin airfoil theory, Eq. (2.4) and Eq. (2.5) provide the expressions for c_l and c_d of an arbitrary two-dimensional thin airfoil at small angle of attack shown in Figure 2.1.

$$c_l = \frac{4\alpha_0}{\sqrt{M_\infty^2 - 1}} \quad (2.4)$$

$$c_d = \frac{4}{\sqrt{M_\infty^2 - 1}} \left[\overline{\left(\frac{dh}{dx}\right)^2} + \alpha_0^2 + \overline{\alpha_c^2(x)} \right] \quad (2.5)$$

It is clear from Eq. (2.4) that the lift of the thin airfoil in supersonic flow simply depends on the mean angle of attack, while the wave drag consists of three parts: drag caused by the angle of attack; drag caused by the camber of the airfoil and drag caused by the thickness of the airfoil.

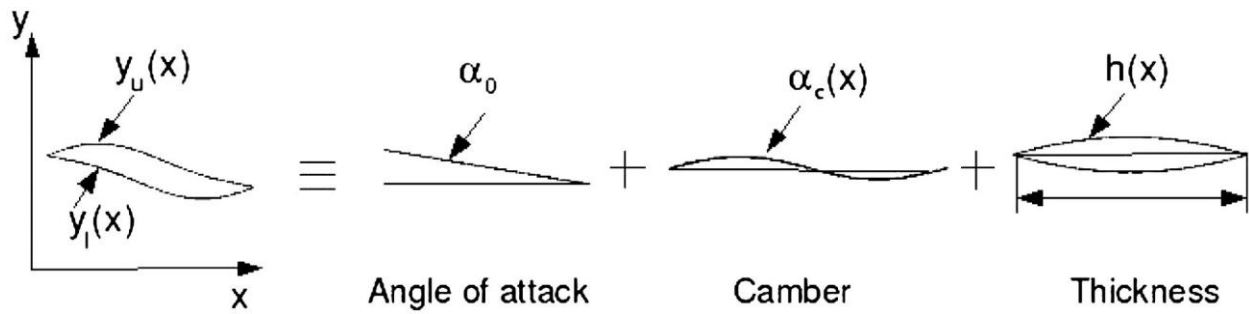


Figure 2.1 Linear decomposition of contributions to lift and drag of an arbitrary thin airfoil at small angle of attack [19]

2.2 Advantageous Effects of Busemann Biplane Airfoil

The biplane airfoil concept proposed by Adolf Busemann [1] is simply to divide a standard diamond-shaped airfoil into two identical components and place the triangular surfaces facing each other at a certain distance apart, as shown in Figure 2.2. The Busemann biplane airfoil design works excellently to reduce the drag due to angle of attack and the drag due to thickness, referred to as the wave reduction effect and the wave cancellation effect respectively.

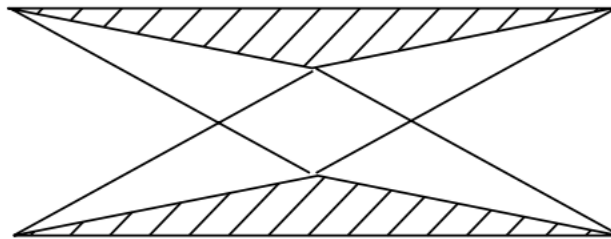


Figure 2.2 Configuration of the Busemann biplane airfoil concept [17]

2.2.1 Wave Reduction Effect

Although the drag due to angle of attack cannot be completely eliminated, it can be significantly reduced by employing the multiple-airfoil concept [7]. To better show this, the general airfoil is simplified to a flat plate airfoil. Thus, c_l is the same as in Eq. 2.4 while c_d can be simplified to

$$c_d = \frac{4\alpha_0^2}{\sqrt{M_\infty^2 - 1}} \quad (2.6)$$

This indicates that the lift coefficient c_l of a general airfoil is proportional to the angle of attack while the drag coefficient c_d of a general airfoil is proportional to the square of the angle of attack.

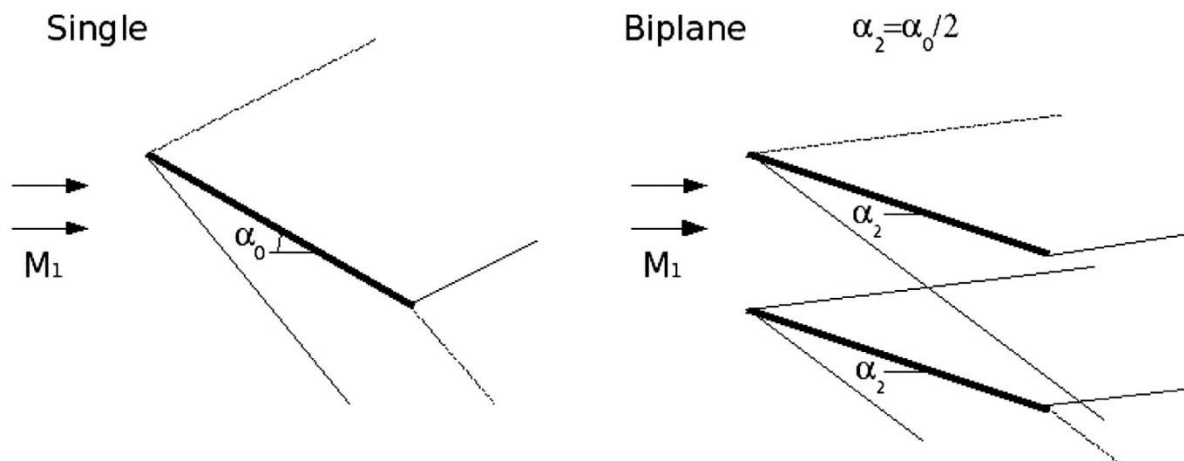


Figure 2.3 Wave reduction effect of the biplane airfoil [17]

In the case of cascade of airfoils which consists of n parallel flat plates with the same chord length c as the single flat plate airfoil shown in Figure 2.3, the angle of attack α_n is set at $\alpha_n = \alpha_0/n$ since the total lift of this n -plate cascade of airfoils c_{ln} is set to be the same as that of the single plate airfoil c_{lS} as described by Eq. (2.7).

$$c_{ln} = n \cdot \frac{4\alpha_n}{\sqrt{M_\infty^2 - 1}} = n \cdot \frac{\frac{4\alpha_0}{n}}{\sqrt{M_\infty^2 - 1}} = c_{ls} \quad (2.7)$$

The wave drag of this n -plate cascade of airfoils c_{dn} equals to the summation of the wave drag of all of the plates:

$$c_{dn} = n \cdot \frac{4\alpha_n^2}{\sqrt{M_\infty^2 - 1}} = n \cdot \frac{4\left(\frac{\alpha_0}{n}\right)^2}{\sqrt{M_\infty^2 - 1}} = \frac{1}{n} \frac{4\alpha_0^2}{\sqrt{M_\infty^2 - 1}} = \frac{1}{n} c_{ds} \quad (2.8)$$

It can be seen from Eq. (2.8) that the n -plate cascade of airfoils produces $\frac{1}{n}$ of the wave drag compared to the single flat plate airfoil while having the same amount of lift (Eq. (2.7)).

In summary, for a combined cascade of multiple-airfoils configuration, the drag caused by the angle of attack or lift effect is reduced since the effective angle of attack or lift effect of each individual airfoil is reduced while the total lift of the combined cascade of multiple-airfoils remains the same. This reduction in wave drag is called the ‘‘Wave Reduction Effect’’ of a combined cascade of multiple-airfoils configuration [7].

2.2.2 Wave Cancellation Effect

The Busemann-type biplane configuration significantly reduces the drag due to thickness as well. By adjusting the distance Δz between the upper and lower components of the airfoil at a specific Mach

number, the strong shock wave generated at the leading edge of one of the components (upper or lower) of the airfoil reaches exactly at the mid-point of the opposite component and is cancelled by the expansion wave generated at that point. Theoretically speaking, the shock wave is completely cancelled, and thus no wave drag due to thickness of the airfoil is produced, as shown in Figure 2.4.

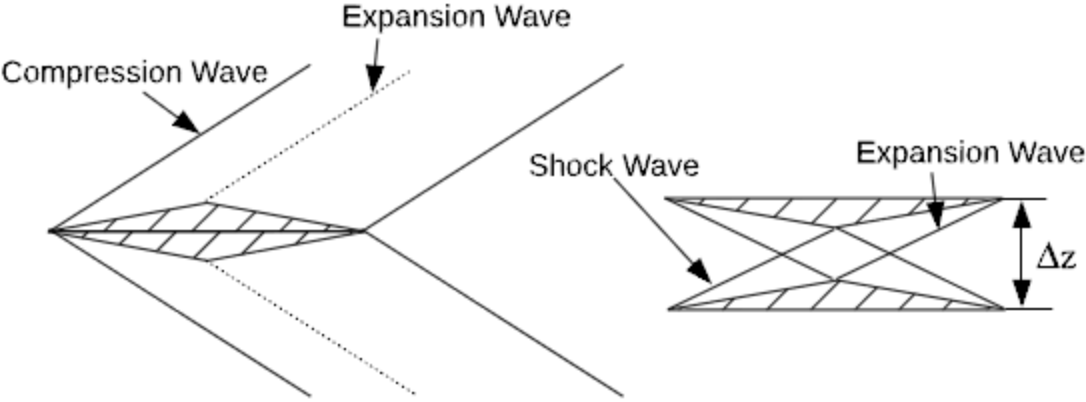


Figure 2.4 Wave cancellation effect [17]

However, because of the entropy increase caused by the shock wave generated between the upper and lower components of the airfoil and due to the nonlinear effects during the shock-expansion process, the wave drag cannot be completely eliminated, and therefore the theoretical zero-drag condition cannot be practically realized [19,20]. At other off-design Mach numbers, the shock wave is cancelled only partially, as shown in Figure 2.5. Thus, the wave drag of the Busemann biplane airfoil does not become zero under off-design conditions.

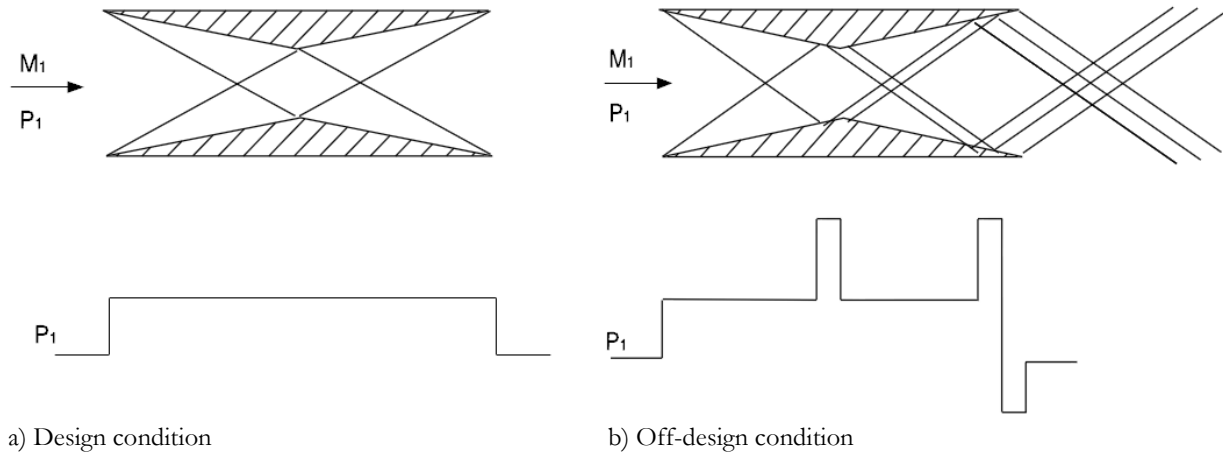


Figure 2.5 Wave formation and pressure changes at design and off-design condition for the Busemann biplane airfoil [17]

2.3 Flow Field Simulation of the Standard Diamond-Shaped Airfoil and the Baseline Busemann Biplane Airfoil under Design and Off-design Conditions

The flow fields of both the standard diamond-shaped airfoil and the Busemann biplane airfoil are computed at zero angle of attack (the zero-lift condition). For meaningful comparison, the total thickness of the two airfoils is set to be equal. For this specific case under consideration, the thickness to chord ratio of the diamond-shaped airfoil is $t/c = 0.1$, while the thickness to chord ratio of the Busemann airfoil is $t/c = 0.05$ for both its upper and lower components. The distance between the upper and lower components of the Busemann airfoil is set to be one half of the chord length to obtain the theoretical minimum drag under the design condition (Mach number $M = 1.7$). The angle of attack is set to be zero since we first consider the non-lifting design and off-design conditions by varying the Mach number below 1.7 and greater than 1.7.

2.3.1 Mesh Generation

The commercial meshing software ANSYS ICEM CFD is used to generate the mesh for computing the flow field. The far field boundary is set at 21-chord-length by 20.5-chord-length rectangle from the center of the airfoil as shown in Figure 2.6. Figure 2.7 and Figure 2.8 show the H mesh configuration generated for the simulations. There are 64 nodes around the two components both horizontally and vertically. In between the two components, the grid has a dimension of 32 nodes (vertically) \times 64 nodes (horizontally). An ICEM replay script file is created to automatically generate the mesh for the flow past different airfoil shapes once it is called.

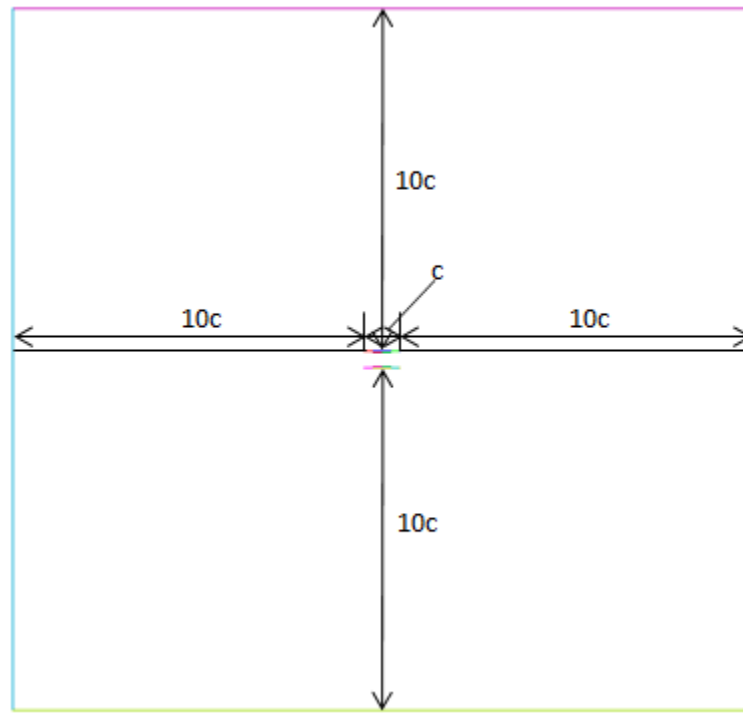


Figure 2.6 Computational domain for the Busemann biplane airfoil configuration

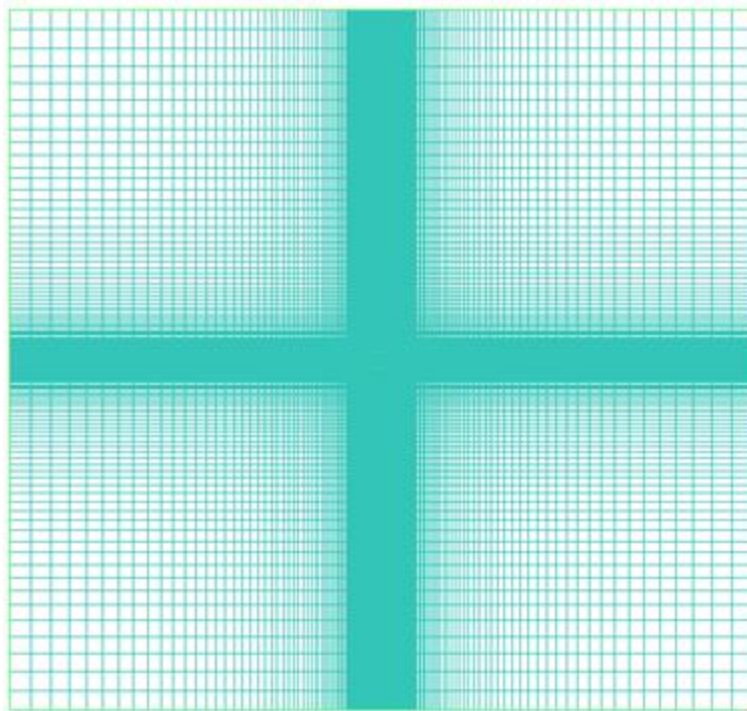


Figure 2.7 H-mesh generated in the computational domain around the Busemann biplane airfoil

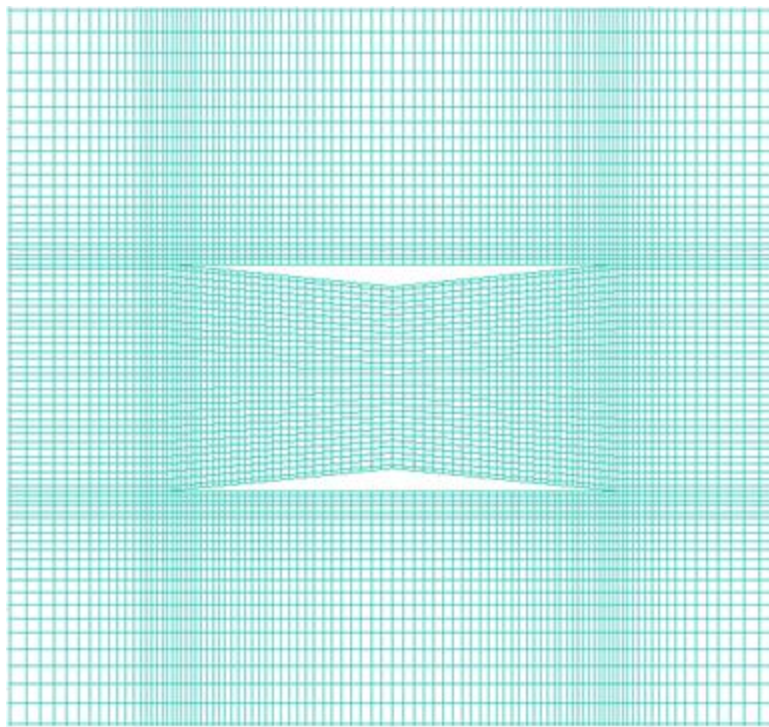


Figure 2.8 Zoomed-in view of H-mesh generated in computational domain around the Busemann biplane airfoil

2.3.2 Steady-State Flow Field Simulation at $M < 1.7$

As has been discussed above, the baseline Busemann biplane airfoil has a much lower drag under its design condition with Mach number $M = 1.7$. However, it has a poor aerodynamic performance under off-design conditions, which may cause much higher drag compared to the standard diamond-shaped airfoil. The configurations of both the standard diamond-shaped airfoil and the baseline Busemann biplane airfoil are shown in Figure 2.9.

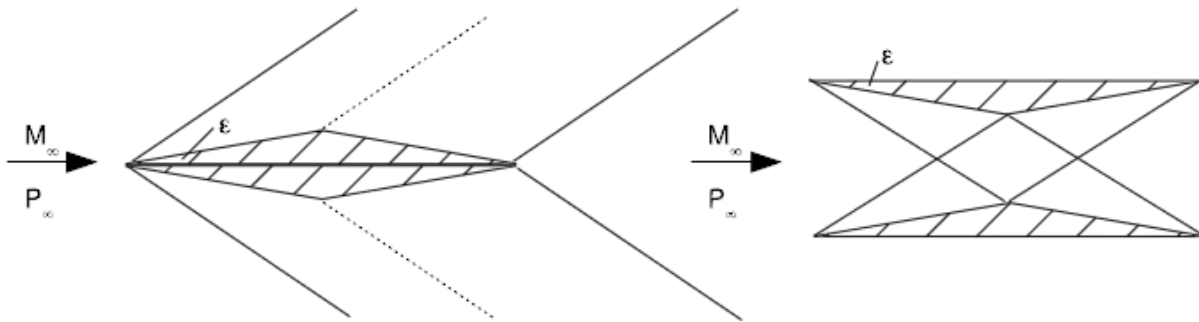


Figure 2.9 Configurations of standard diamond-shaped airfoil (left) and baseline Busemann biplane airfoil (right)

Figure 2.10 shows the drag coefficient c_d of both the standard diamond-shaped airfoil and the baseline Busemann biplane airfoil under zero-lift condition over a range of Mach numbers ($0.3 \leq M \leq 3.3$). The simulations are performed using ANSYS FLUENT; the flow field is initialized with an impulsive uniform flow.

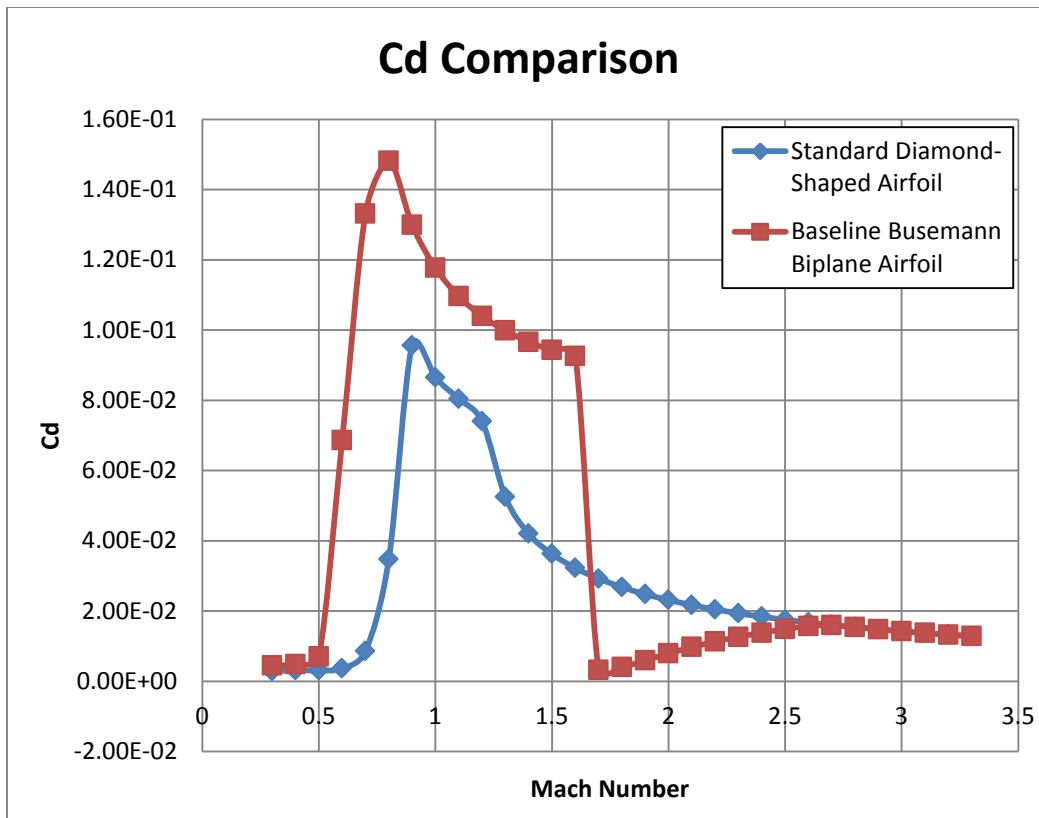


Figure 2.10 c_d comparison of two different airfoils under non-lifting condition

As shown in Figure 2.10, the baseline Busemann biplane airfoil has higher drag compared to the standard diamond-shaped airfoil when the Mach number is low ($0.3 \leq M \leq 1.6$). In the range $1.7 \leq M \leq 2.7$ however, the drag generated by the baseline Busemann biplane airfoil is smaller than that of the standard diamond-shaped airfoil, especially at $M = 1.7$ which is the design condition for the Busemann airfoil. Table 2.1 shows a detailed comparison of the drag coefficient c_d of these two airfoils for the range of Mach numbers $0.3 \leq M \leq 3.3$.

Table 2.1 Drag coefficient c_d for diamond-shaped airfoil and Busemann airfoil over a range of Mach numbers

M	Cd (Diamond)	Cd (Busemann)
0.3	2.83E-03	4.45E-03
0.4	3.15E-03	4.87E-03
0.5	3.09E-03	7.01E-03
0.6	3.63E-03	6.86E-02
0.7	8.54E-03	1.33E-01
0.8	3.47E-02	1.48E-01
0.9	9.56E-02	1.30E-01
1	8.65E-02	1.18E-01
1.1	8.04E-02	1.10E-01
1.2	7.40E-02	1.04E-01
1.3	5.25E-02	9.98E-02
1.4	4.20E-02	9.66E-02
1.5	3.62E-02	9.43E-02
1.6	3.22E-02	9.26E-02
1.7	2.92E-02	3.21E-03
1.8	2.68E-02	3.99E-03
1.9	2.48E-02	5.94E-03
2	2.31E-02	7.94E-03
2.1	2.17E-02	9.73E-03
2.2	2.05E-02	1.13E-02
2.3	1.94E-02	1.26E-02
2.4	1.84E-02	1.38E-02
2.5	1.75E-02	1.48E-02
2.6	1.67E-02	1.57E-02
2.7	1.60E-02	1.59E-02
2.8	1.54E-02	1.54E-02
2.9	1.48E-02	1.48E-02
3	1.42E-02	1.42E-02
3.1	1.37E-02	1.37E-02
3.2	1.33E-02	1.33E-02
3.3	1.29E-02	1.28E-02

Both Figure 2.10 and Table 2.1 demonstrate the advantage of the Busemann biplane airfoil since it produces much lower drag near its design condition ($M = 1.7$) due to the wave reduction and wave cancellation effects. The simulations shown above demonstrate that the baseline Busemann biplane airfoil has a very good performance at its design condition ($M = 1.7$), while it has much higher drag at off-design conditions due to the choked-flow phenomenon at lower Mach numbers ($M < 1.7$).

Figure 2.11 through Figure 2.14 show the flow field around the baseline Busemann biplane airfoil under two different off-design conditions both of which result in high drag but due to different reasons. At $M = 0.8$, as shown in Figure 2.11, the flow between the two airfoil components reaches Mach one at the midchord of the airfoil and then increases further downstream to supersonic. After the trailing edge of the airfoil however, the flow speed again drops to subsonic and hence forms a vertical shock wave. The supersonic flow in the rear part of the airfoil (after the mid-point) creates the low-pressure region, as shown in Figure 2.12, which leads to higher wave drag ($c_d = 0.148$) at this off-design condition of $M = 0.8$. At $M = 1.6$ however, as shown in Figure 2.13, the flow field is very different from the $M = 0.8$ subsonic condition discussed above. A bow shock is formed in front of the leading edge of the airfoil. The flow speed drops to subsonic and a high-pressure region, as shown in Figure 2.14, is created which again generates a high drag ($c_d = 0.0926$).

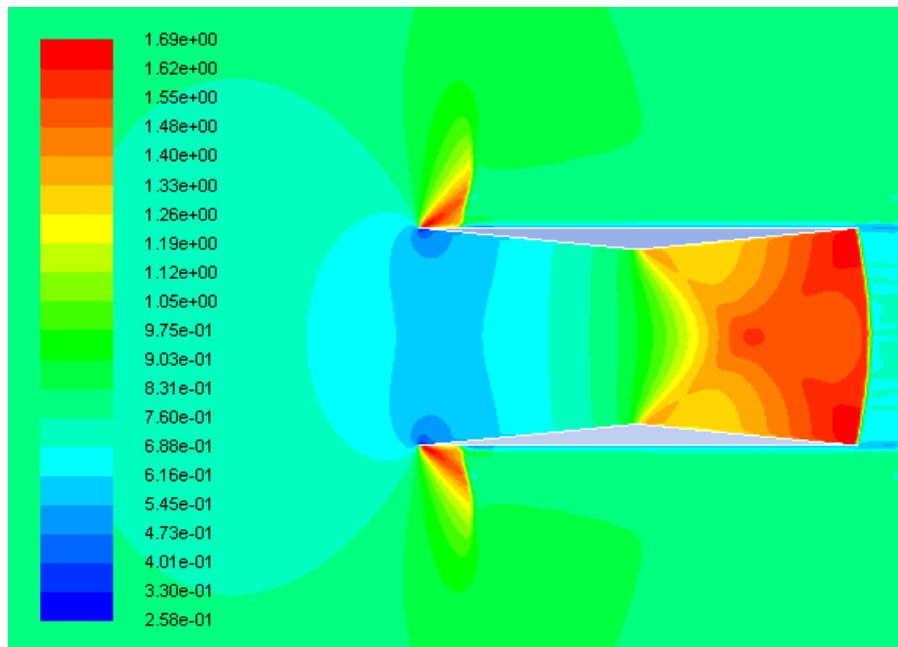


Figure 2.11 Velocity contours around the Busemann airfoil at $M = 0.8$

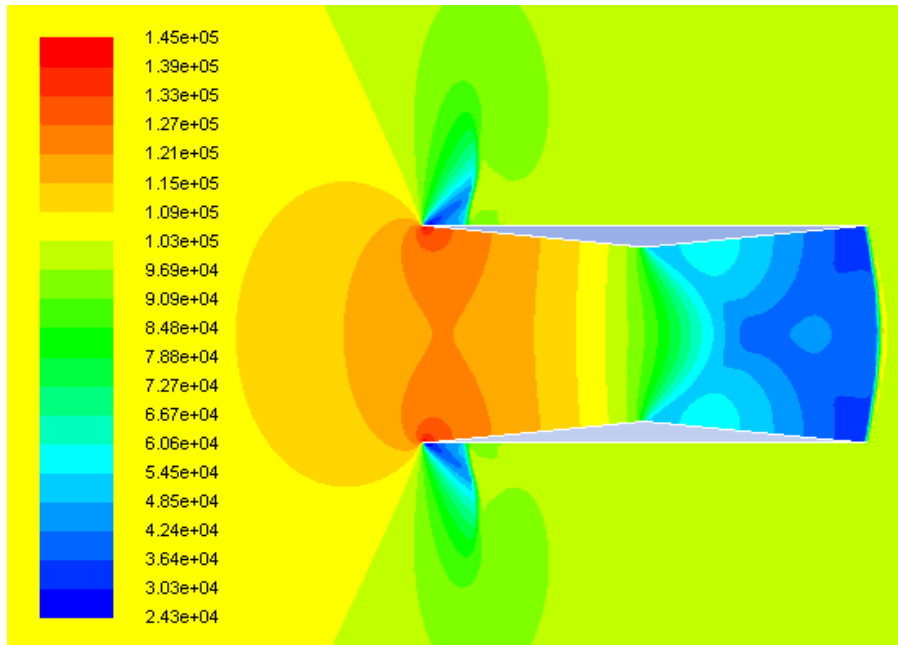


Figure 2.12 Pressure contours around the Busemann airfoil at $M = 0.8$

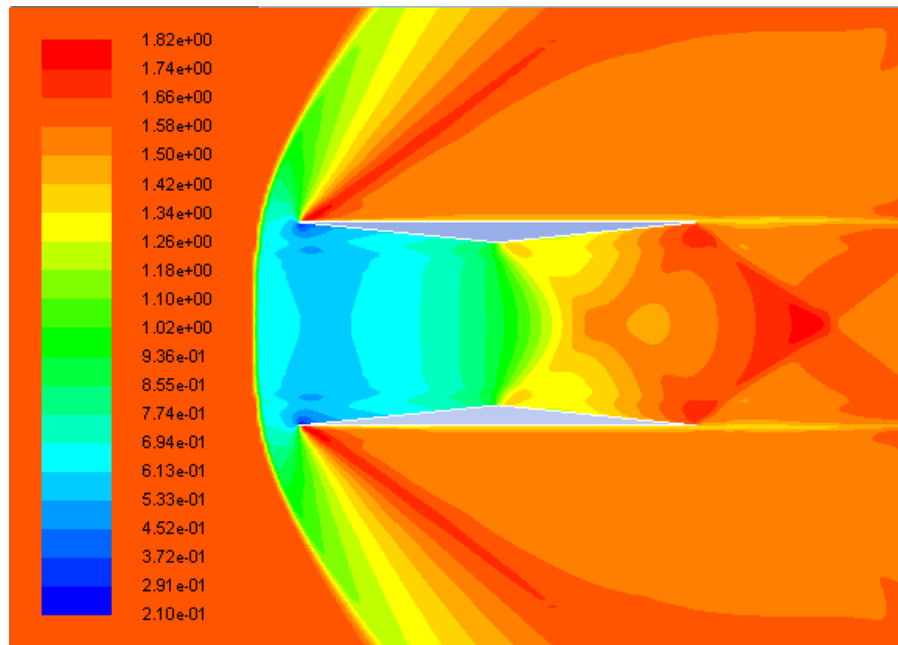


Figure 2.13 Velocity contours around the Busemann airfoil at $M = 1.6$

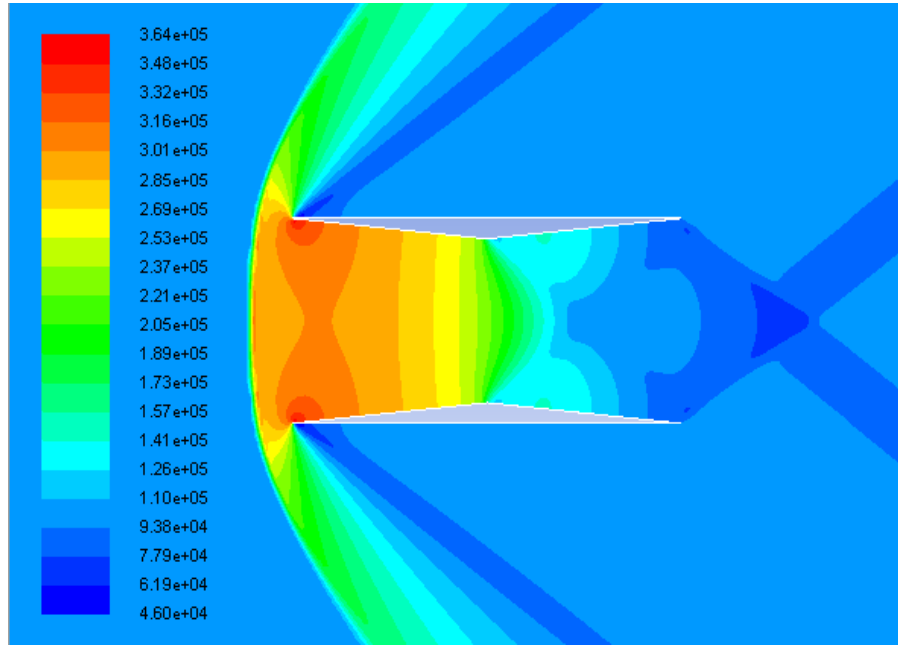


Figure 2.14 Pressure contours around the Busemann airfoil at $M = 1.6$

2.3.3 Flow Field Simulation during Acceleration and Deceleration

Besides the high drag been generated under off-design conditions, there is an even worse problem caused by the flow-hysteresis phenomenon during acceleration and choked-flow phenomenon during deceleration that need to be addressed under off-design conditions. To demonstrate these phenomena, flow field simulations are conducted both under acceleration and deceleration using the previous simulation results shown in Figure 2.10 as the initial condition. Figure 2.15 shows two separated c_d curves during acceleration and deceleration of the biplane airfoil. As been shown in Figure 2.15, the change in acceleration and deceleration c_d curves occurs in the range $M = 1.6$ to $M = 2.1$, where the blue dash line represents the c_d of the Busemann biplane airfoil during acceleration and the red solid line represents the c_d of the Busemann biplane airfoil during deceleration. This separation between the two c_d curves is caused by the flow-hysteresis phenomenon during acceleration and the choked-flow phenomenon during deceleration. Therefore,

in order to minimize the difference in the drag coefficients of the Busemann biplane airfoil during acceleration and deceleration as well as to significantly decrease the drag, shape optimization of the Busemann airfoil is conducted using SOGA to eliminate the flow-hysteresis phenomenon during acceleration and the choked-flow phenomenon during deceleration; it is discussed in Chapter 3. Table 2.2 compares the c_d for diamond-shaped airfoil and Busemann airfoil under acceleration and deceleration at various Mach numbers.

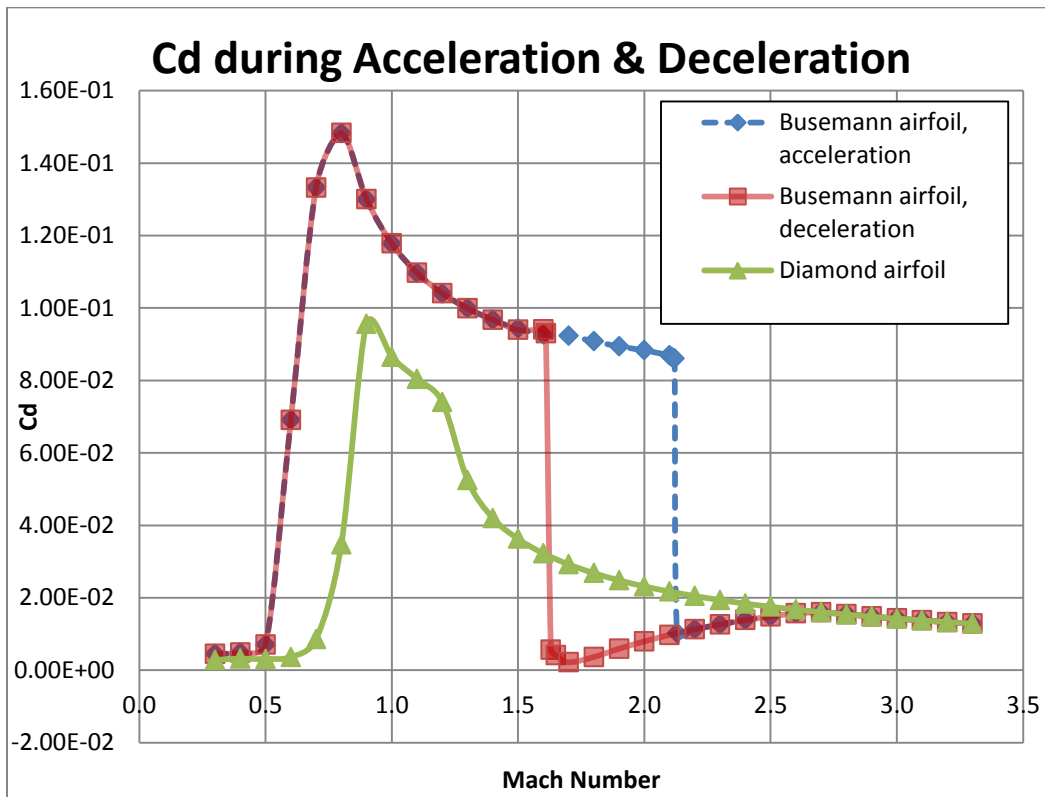


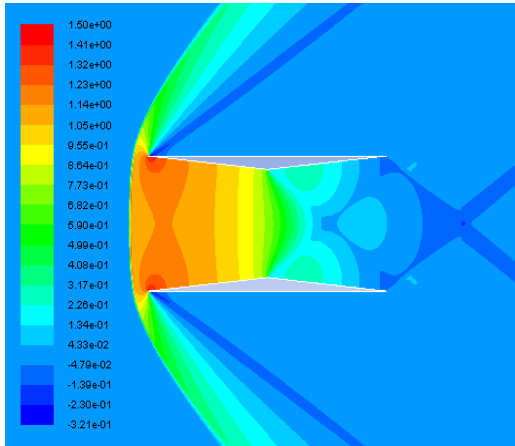
Figure 2.15 c_d plots for diamond-shaped airfoil and Busemann airfoil during acceleration and deceleration under non-lifting condition

2.3.3.1 Flow Field of Busemann Biplane Airfoil during Acceleration

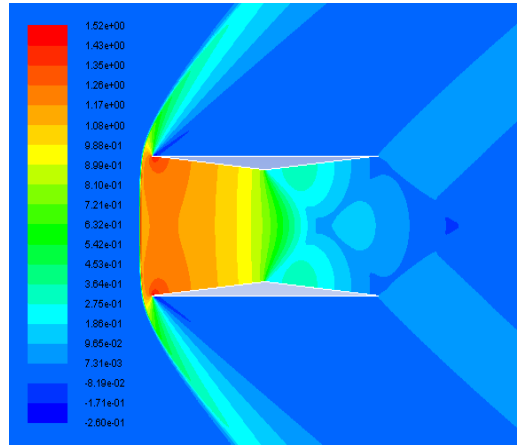
In this section, the flow-hysteresis phenomenon during acceleration is examined for the baseline Busemann biplane airfoil under non-lifting condition. The pressure coefficient contours for flow past the Busemann airfoil during acceleration are shown in Figure 2.16. To simulate the non-lifting condition, the angle of attack is set to zero. The resulting non-lifting flow field around the Busemann airfoil is shown in Figure 2.16 at various supersonic Mach numbers ranging from $M = 1.7$ to $M = 2.13$. As shown in Figure 2.16, the bow shock exists in front of the Busemann airfoil and does not disappear until the Mach number reaches 2.13. It can also be noticed that there is a subsonic region behind the bow shock between the upper and lower components of the airfoil where the pressure coefficients are high. The result of the presence of this bow shock in front of the airfoil is a substantial increase in drag compared to that at the design condition. However, when the Mach number increases from 2.12 to 2.13, the bow shock is swallowed into the upper and lower components of the Busemann airfoil and is replaced by two oblique shock waves, and the subsonic region between the two airfoil components finally disappears as shown in Figure 2.16(f). The drag coefficient of the airfoil also decreases dramatically and the flow past the airfoil develops into a state similar to that under the design condition. Figure 2.16 illustrates the poor performance of the baseline Busemann airfoil under off-design conditions. During acceleration, the design condition cannot be achieved at Mach number 1.7. And the drag coefficient at Mach number 1.7 is much higher ($c_d = 0.0923$) compared to the standard diamond-shaped airfoil ($c_d = 0.0292$). Furthermore due to the flow-hysteresis phenomenon, the drag coefficients c_d during acceleration and deceleration are different as shown in Figure 2.15 and Table 2.2.

Table 2.2 Drag Coefficients c_d for Busemann airfoil and standard diamond airfoil at various Mach numbers

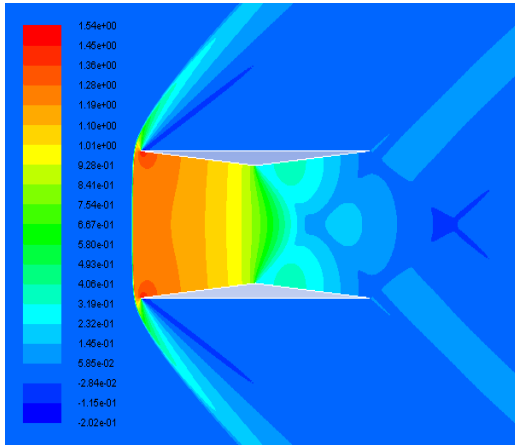
Busemann Acceleration		Busemann Deceleration		Standard Diamond	
M	Cd	M	Cd	M	Cd
0.3	4.45E-03	0.3	4.45E-03	0.3	2.83E-03
0.4	4.87E-03	0.4	4.87E-03	0.4	3.15E-03
0.5	7.01E-03	0.5	7.01E-03	0.5	3.09E-03
0.6	6.91E-02	0.6	6.91E-02	0.6	3.63E-03
0.7	1.33E-01	0.7	1.33E-01	0.7	8.54E-03
0.8	1.48E-01	0.8	1.48E-01	0.8	3.47E-02
0.9	1.30E-01	0.9	1.30E-01	0.9	9.56E-02
1.0	1.18E-01	1	1.18E-01	1	8.65E-02
1.1	1.10E-01	1.1	1.10E-01	1.1	8.04E-02
1.2	1.04E-01	1.2	1.04E-01	1.2	7.40E-02
1.3	9.99E-02	1.3	9.99E-02	1.3	5.25E-02
1.4	9.67E-02	1.4	9.67E-02	1.4	4.20E-02
1.5	9.43E-02	1.5	9.40E-02	1.5	3.62E-02
1.6	9.26E-02	1.6	9.40E-02	1.6	3.22E-02
1.7	9.23E-02	1.61	9.30E-02	1.62	3.16E-02
1.8	9.08E-02	1.63	5.59E-03	1.7	2.92E-02
1.9	8.94E-02	1.65	4.14E-03	1.8	2.68E-02
2.0	8.83E-02	1.7	2.18E-03	1.9	2.48E-02
2.10	8.69E-02	1.8	3.64E-03	2	2.31E-02
2.11	8.64E-02	1.9	5.94E-03	2.1	2.17E-02
2.12	8.60E-02	2	7.94E-03	2.2	2.05E-02
2.13	1.03E-02	2.1	9.73E-03	2.3	1.94E-02
2.2	1.13E-02	2.2	1.13E-02	2.4	1.84E-02
2.3	1.26E-02	2.3	1.26E-02	2.5	1.75E-02
2.4	1.38E-02	2.4	1.38E-02	2.6	1.67E-02
2.5	1.48E-02	2.5	1.48E-02	2.7	1.60E-02
2.6	1.57E-02	2.6	1.57E-02	2.8	1.54E-02
2.7	1.59E-02	2.7	1.59E-02	2.9	1.48E-02
2.8	1.54E-02	2.8	1.54E-02	3	1.42E-02
2.9	1.48E-02	2.9	1.48E-02	3.1	1.37E-02
3.0	1.42E-02	3.0	1.42E-02	3.2	1.33E-02
3.1	1.37E-02	3.1	1.37E-02	3.3	1.29E-02
3.2	1.33E-02	3.2	1.33E-02		
3.3	1.28E-02	3.3	1.28E-02		



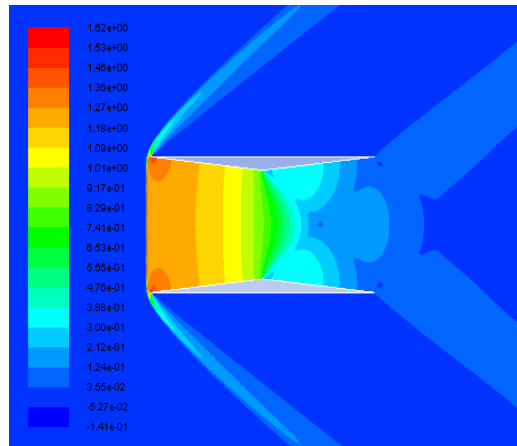
a) $M_\infty = 1.7, c_d = 0.0923$



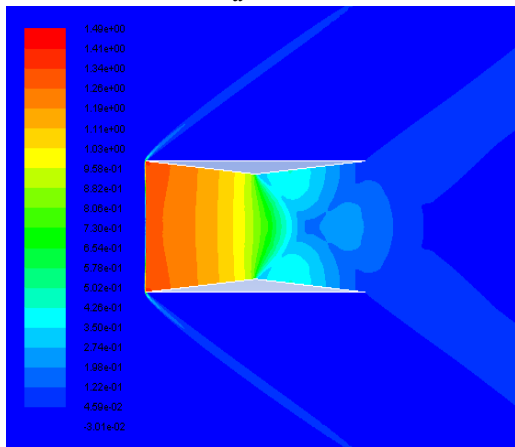
b) $M_\infty = 1.8, c_d = 0.0908$



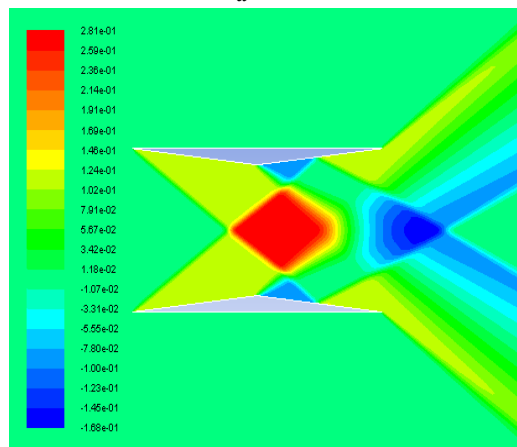
c) $M_\infty = 1.9, c_d = 0.08939$



d) $M_\infty = 2.0, c_d = 0.0883$



e) $M_\infty = 2.12, c_d = 0.0860$



f) $M_\infty = 2.13, c_d = 0.010264$

Figure 2.16 c_p -contours around the Busemann biplane airfoil with zero-lift during acceleration

2.3.3.2 Flow Field of Busemann Biplane Airfoil during Deceleration

In this section, the choked-flow phenomenon of the baseline Busemann biplane airfoil during deceleration is examined. The contours of pressure coefficients for flow past the Busemann airfoil during deceleration are shown in Figure 2.17. To simulate the non-lifting condition, the angle of attack is set to zero as before. The resulting non-lifting flow field around the Busemann airfoil is shown in Figure 2.17 at various supersonic Mach numbers ranging from $M = 1.8$ to $M = 1.62$. As shown in Figure 2.17, a different flow field appears within the small range near the design Mach number ($M = 1.7$); a relatively high drag coefficient still occurs as the Mach number further decreases during deceleration. A strong bow shock is formed in front of the airfoil when the Mach number drops from 1.63 to 1.62, while the drag coefficient increases dramatically from 0.005594 to 0.0926, which is substantially higher than that of the standard diamond-shaped airfoil ($c_d = 0.03158$). The flow between the two components of the airfoil is choked at the location of the maximum thickness of the Busemann airfoil and a subsonic region is formed. This is also a clear indication of the poor performance of the baseline Busemann biplane airfoil at off-design conditions since the drag coefficient of the Busemann airfoil is much higher than that of the standard diamond-shaped airfoil for $M \leq 1.62$.

In conclusion, the baseline Busemann biplane airfoil produces a substantially higher drag in the low Mach number range (below the design Mach number of $M = 1.7$). Additionally, it is necessary to accelerate the Busemann biplane airfoil to a much higher Mach number ($M = 2.13$) to reach the shockwave-swallowing state, while producing dramatically higher drag and largely decreasing its efficiency, and then decelerate to a lower velocity to achieve the design condition at Mach number

$M = 1.7$. As a result, the baseline Busemann biplane airfoil design needs to be modified and optimized to avoid or at least reduce the high drag coefficient caused by the flow-hysteresis phenomenon during acceleration and the choked-flow phenomenon during deceleration.

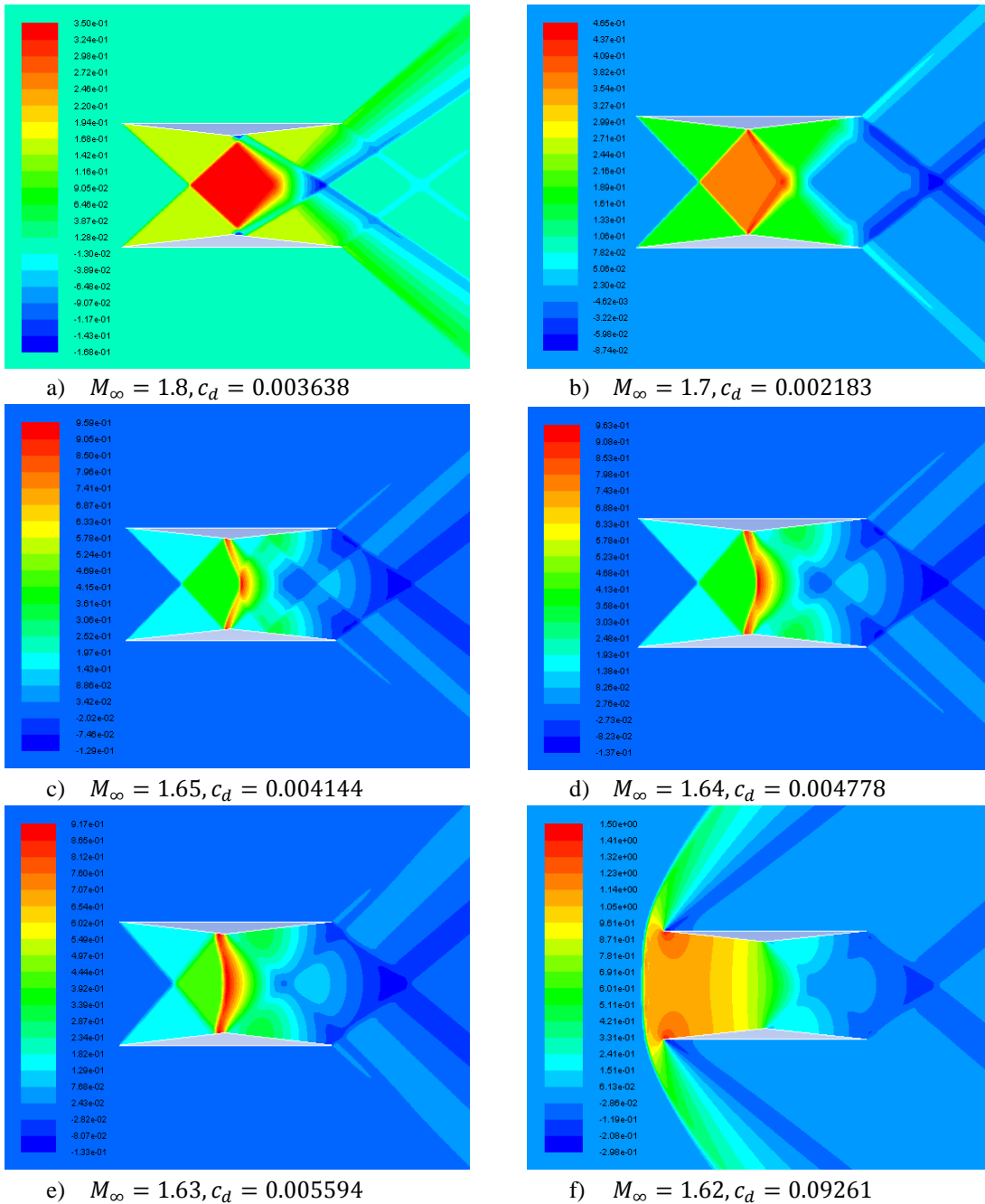


Figure 2.17 c_p -contours around the Busemann biplane airfoil with zero-lift during deceleration

Chapter 3

Optimization of Busemann-Type Biplane Airfoil

In this chapter, the shape optimization procedure for the baseline Busemann biplane airfoil using the genetic algorithms (GA) and the results of the optimized Busemann-type airfoil under both non-lifting and lifting conditions are presented.

3.1 Genetic-Algorithm-Based Optimization Procedure

The optimization process is established by coupling a Single-Objective Genetic Algorithm (SOGA) or a Multi-Objective Genetic Algorithm (MOGA) based optimization method with the mesh generation software ANSYS-ICEM and the CFD solver ANSYS-FLUENT as shown in Fig. 3.1.

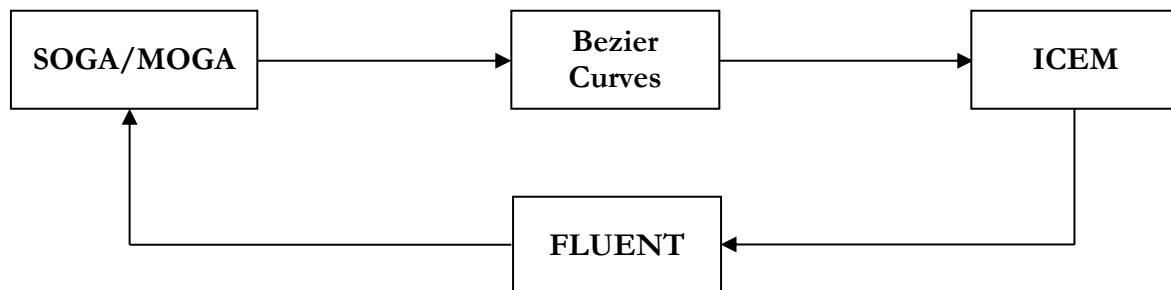


Figure 3.1 Schematic of information flow in the optimization process

Every individual (airfoil) in each generation of SOGA/MOGA is represented by a set of control points, which randomly generate the airfoil shape by using the Bezier Curves. The mesh generation software ICEM is used to generate a two-dimensional structured mesh around the airfoil as an input to the CFD solver FLUENT, which is then used to calculate the supersonic inviscid flow fields for specific flow conditions. Based on the fitness values of all airfoil shapes in a given generation, SOGA/MOGA is applied to create the next generation of airfoils and this whole process is repeated until the optimal fitness value is obtained. The airfoil shape that corresponds to the optimal fitness value is the final shape of the optimized airfoil [18].

3.1.1 Overview of Genetic Algorithm

Genetic algorithms (GAs) are a class of stochastic optimization algorithms inspired by the biological evolution [20]. They efficiently exploit historical information to speculate on new offspring with improved performance [21]. For the Busemann biplane airfoil in particular, genetic algorithm is used to generate new shapes that produce much lower drag by minimizing the flow-hysteresis phenomenon during acceleration and the choked-flow phenomenon during deceleration. Generally, The GA employs the following steps to complete the optimization process:

1. Initialization: Randomly generates a group of individuals.
2. Evaluation: Evaluates the fitness of each individual generated.
3. Natural selection: Individuals that have the lowest fitness are removed.
4. Reproduction: Pairs of the individuals are picked to produce the offspring, which is often done by roulette wheel sampling. A crossover function is then used to produce the offspring.
5. Mutation: Randomly modifies some small percentage of the population.

6. Check for convergence: If the current generation has converged, the individual having the best fitness will be returned. Otherwise, the process will be repeated starting from step 2 until the pre-defined tolerance criteria for acceptable change in fitness between generations is met.

3.2 Shape Optimization of Busemann Biplane Airfoil under Nonlifting Condition

3.2.1 Application of Single-Objective Genetic Algorithm (SOGA)

For the non-lifting condition, where the lift coefficient $c_l = 0$, a Single-Objective Genetic Algorithm (SOGA) is employed for the shape optimization of the Busemann biplane airfoil. The single objective to be achieved is to minimize the drag coefficient c_d . Multiple design points (Mach numbers) are used during the optimization process due to the flow-hysteresis and choked-flow phenomena caused by the baseline Busemann biplane airfoil during acceleration and deceleration respectively. For the multi-point optimization, the fitness function employed is a weighted average of the drag coefficients c_d , which can be written in the form:

$$I = \frac{\sum_{i=1}^n w_i I_i}{\sum_{i=1}^n w_i} \quad (3.1)$$

where I is the weighted average drag coefficient and “ i ” denotes a design point (related to Mach number). For the optimization process for nonlifting condition, an evenly weighted average of drag coefficients c_d at different Mach numbers is employed. Therefore Eq. (3.1) reduces to

$$I = \frac{\sum_{i=1}^n I_i}{n} \quad (3.2)$$

Since the wave drag of the Busemann biplane airfoil is much lower when the flow is unchoked, it is highly desirable that the strong bow shock wave in front of the airfoil is swallowed into the area between the upper and lower components of the airfoil before the flow speed approaches the design Mach number ($M = 1.7$). Although a higher weight can be assigned to the most important design Mach number ($M = 1.7$) to produce a lower drag at that design point, previous research conducted by Hu et al. [17] has shown that a slightly higher drag coefficient c_d is obtained at lower Mach numbers during acceleration and deceleration, as given in Table 3.1 and Table 3.2 respectively if such an uneven weighting method is used.

Table 3.1 c_d comparison at zero lift condition during acceleration (1 count = 0.0001) [17]

Mach number	1.1	1.2	1.3	1.4	1.5	1.6	1.7
Baseline	1050	996	957	928	906	889	873
Optimized (even)	527	473	419	376	332	112	106
Optimized (uneven)	539	486	428	382	336	107	101

Table 3.2 c_d comparison at zero lift condition during deceleration (1 count = 0.0001) [17]

Mach number	1.7	1.6	1.5	1.4	1.3	1.2	1.1
Baseline	873	889	906	928	957	996	1050
Optimized (even)	106	112	127	152	419	473	527
Optimized (uneven)	101	107	122	146	428	486	539

Here, we have a total of seven design points in the Mach numbers range $M = 1.1$ to $M = 1.7$, for both acceleration and deceleration. The reason for picking these seven design points is that this

range of Mach numbers, $M = 1.1$ to $M = 1.7$, provides the critical region before the Mach number increases to the design condition $M = 1.7$; it is the region in which we want to keep the drag coefficient as low as possible. Both acceleration and deceleration scenarios are considered in order to reduce the flow-hysteresis phenomenon and the choked flow phenomenon respectively.

A code in MATLAB package is developed and utilized in the optimization process of the airfoil as well as for ICEM meshing and FLUENT flow field calculations. All SOGA parameters are defined based on the GA methodology, as shown in Table 3.3.

Table 3.3 GA parameters for shape optimization of Busemann biplane airfoil under nonlifting condition

GA Parameters	Description
Generation Size	8 individuals per generation
Number of Generations	Maximum of 50 generations if convergence not obtained
Number of Design Variables	14 in total, 7 for acceleration and 7 for deceleration
Selection Type	Roulette Wheel Selection
Crossover Rate	0.7
Mutation Rate	0.1
Error of Mutation Constant	0.8, which determines how much mutation affects the curves as generations proceed

3.2.2 Airfoil Parameterization

The shape of the airfoil is generated by using the Bezier curves (third-order polynomials). Bezier curves are frequently used in computer graphics to produce curves which appear to be reasonably smooth at all scales. One of the main reasons that Bezier curves are used in computer graphics is that they can be efficiently constructed; each Bezier curve is simply defined by a set of control points [22].

For the non-lifting case in specific, with $c_l = 0$, since the upper and lower components of the airfoil are symmetric with respect to the horizontal axis, only the upper component of the airfoil needs to be defined; the lower component is the mirror image with respect to the horizontal axis. For the upper component of the airfoil, there are two lines that need to be drawn to define its shape. Since the thickness distribution for the entire airfoil remains the same as that of the baseline Busemann biplane airfoil, the y-coordinates of the lower line are defined by Eq. (3.3) and Eq. (3.4). Thus the upper line (which is a straight horizontal line for the baseline Busemann airfoil) is the only line that needs to be generated by the Bezier curves in order to define the shape of the whole biplane airfoil.

$$y_{low} = y_{up} - 0.05 - 0.1 \times x_{up} \quad (-0.5 \leq x_{up} \leq 0) \quad (3.3)$$

$$y_{low} = y_{up} - 0.05 + 0.1 \times x_{up} \quad (0 \leq x_{up} \leq 0.5) \quad (3.4)$$

In Eq. (3.3) and Eq. (3.4), the subscripts ‘low’ and ‘up’ correspond to the lower and upper line of the airfoil; the origin is at the center of the upper line. Two Bezier curves are used to generate the shape-defining upper line. Each Bezier curve is defined by a set of four control points. Each control point is constrained by a specified range of x- and y-coordinates as shown in Table 3.4. Figure 3.2 shows a randomly generated Busemann-type biplane airfoil shape using Bezier curves. Figure 3.3 and Figure 3.4 show the detailed geometry of Bezier curve #1 and Bezier curve #2 along with their control points.

Table 3.4 Coordinate range for each control point of the Bezier curves for the airfoil under nonlifting condition

	Coordinate	Upper Limit	Lower Limit
Bezier curve #1	x1	Fixed at x=-0.5	
	x2	-0.05	-0.5
	x3	Fixed at x=-0.05	
	x4	Fixed at x=0	
	y1	0.25	0.2
	y2	0.25	0.2
	y3	Fixed at y=0.25	
	y4	Fixed at y=0.25	
Bezier curve #2	x1	Fixed at x=0	
	x2	Fixed at x=0.05	
	x3	0.5	0.05
	x4	Fixed at x=0.5	
	y1	Fixed at y=0.25	
	y2	Fixed at y=0.25	
	y3	0.25	0.2
	y4	0.25	0.2

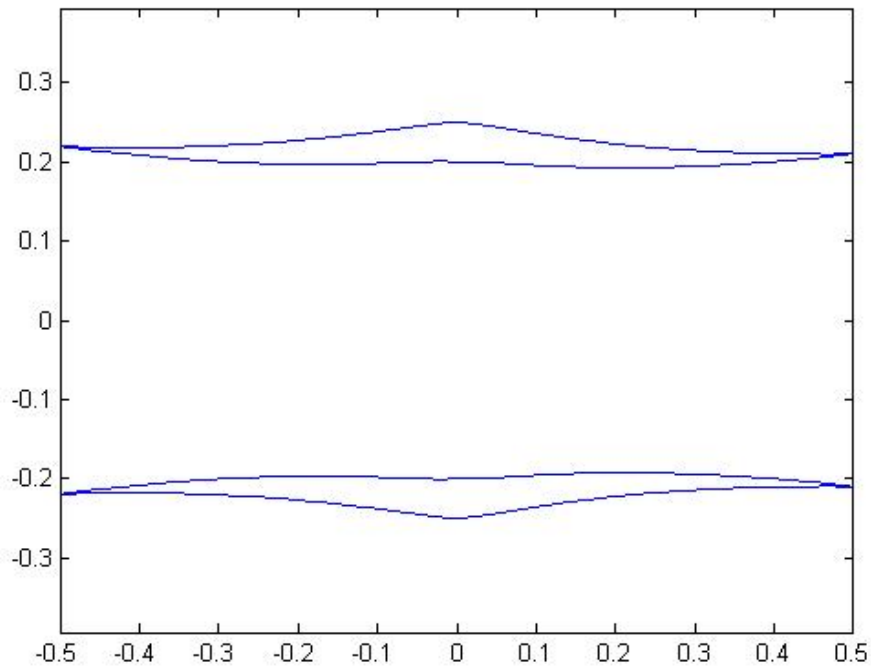


Figure 3.2 Randomly generated Busemann-type biplane airfoil shape using Bezier curves

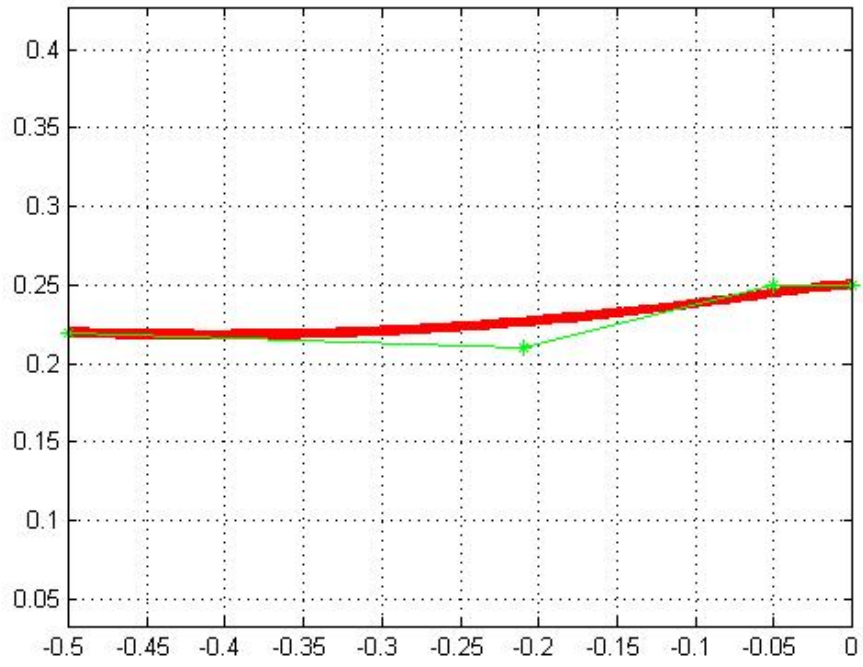


Figure 3.3 Detailed geometry of Bezier curve #1 and its control points

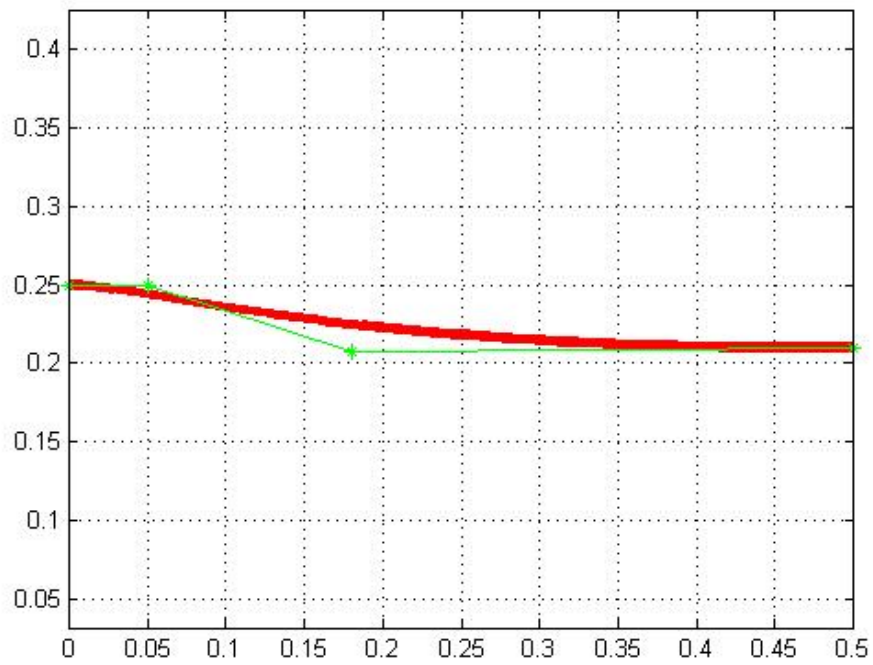


Figure 3.4 Detailed geometry of Bezier curve #2 and its control points

3.2.3 Optimization Results

After implementing SOGA for 20 generations with 8 individuals in each generation, an optimal shape result for symmetric Busemann-type biplane airfoil under non-lifting condition with minimum drag is obtained. Figure 3.5 shows the geometry of the original Busemann biplane airfoil (red) and the optimized Busemann biplane airfoil (blue) under nonlifting condition. Figure 3.6 and Figure 3.7 show the two final Bezier curves used to generate the optimized Busemann biplane airfoil shape along with their control points.

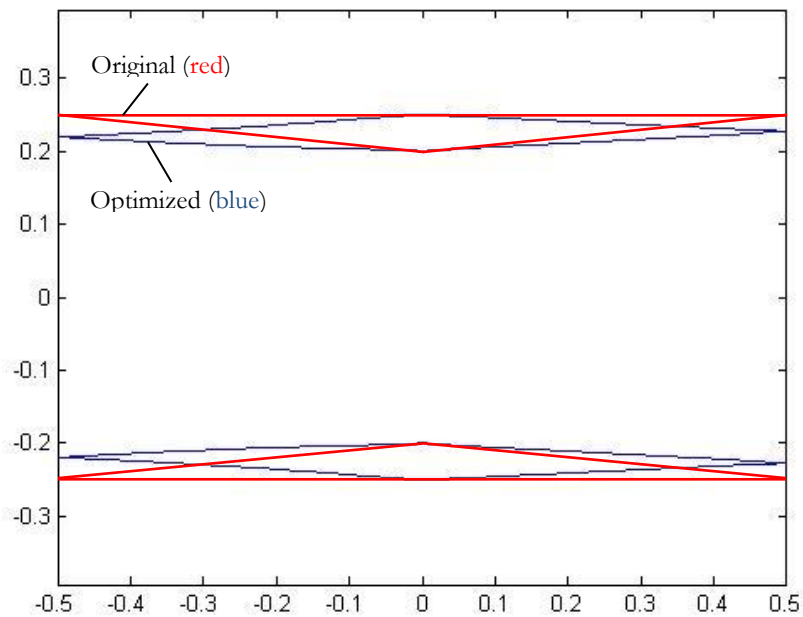


Figure 3.5 Geometry of both the original and optimized Busemann biplane airfoil under nonlifting condition

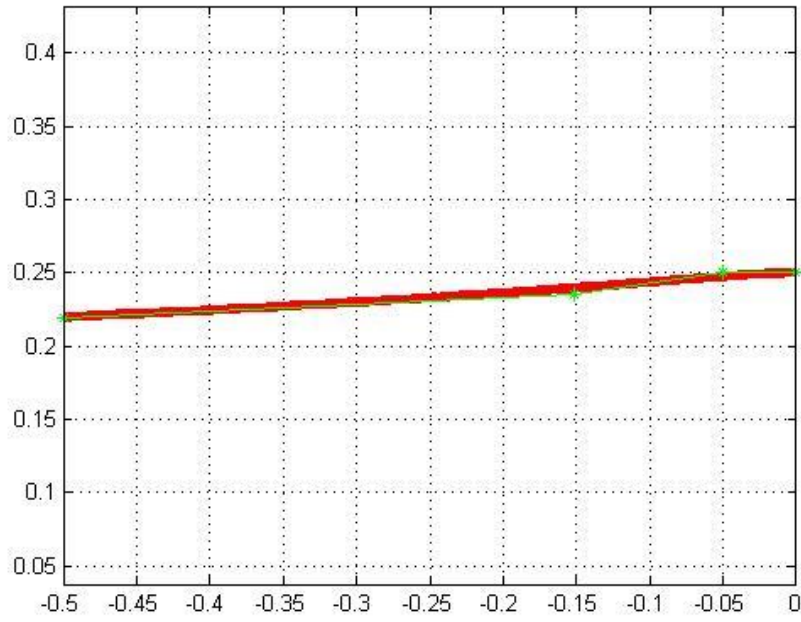


Figure 3.6 Bezier curve #1 and its control points for the optimized Busemann biplane airfoil under nonlifting condition

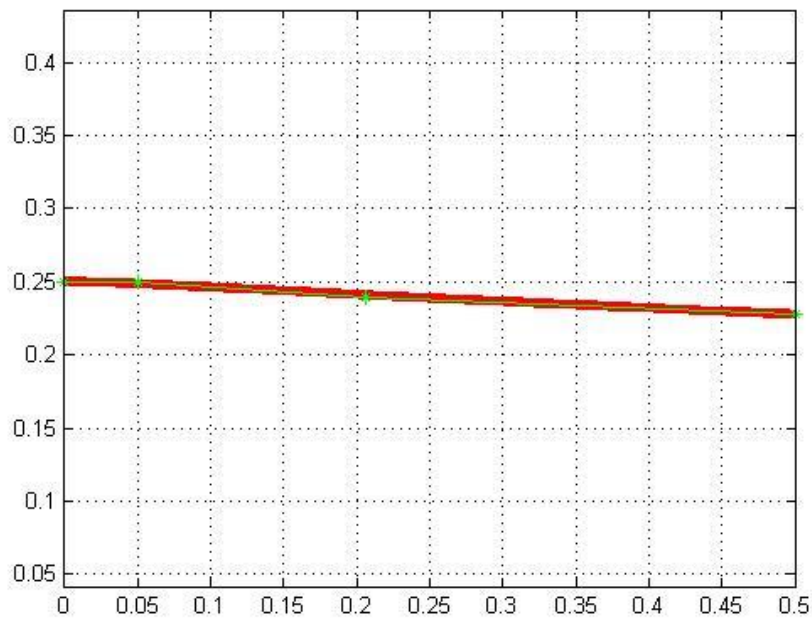


Figure 3.7 Bezier curve #2 and its control points for the optimized Busemann biplane airfoil under nonlifting condition

The drag coefficients for the seven design points are compared in Table 3.5 and Table 3.6 for both the original and optimized Busemann biplane airfoil under nonlifting condition during acceleration and deceleration respectively. As shown in Table 3.5 and Table 3.6, the baseline Busemann biplane airfoil is choked at all Mach numbers within the optimization range; while the optimized Busemann biplane airfoil unchokes at $M = 1.5$ during acceleration and chokes at $M = 1.3$ during deceleration. Even under choked conditions during both acceleration and deceleration, the optimized Busemann biplane airfoil has significantly lower drag compared to the baseline Busemann biplane airfoil. The only point where the optimized Busemann biplane airfoil has a higher drag compared to the original airfoil is at $M = 1.7$ during deceleration.

Table 3.5 c_d comparison for the original and optimized Busemann biplane airfoil at nonlifting condition during acceleration (1 count = 0.0001)

Mach Number	1.1	1.2	1.3	1.4	1.5	1.6	1.7
Baseline	1097	1040	999	967	943	926	923
Optimized (GA)	622	506	428	389	132	114	104

Table 3.6 c_d comparison for the original and optimized Busemann biplane airfoil at nonlifting condition during deceleration (1 count = 0.0001)

Mach Number	1.7	1.6	1.5	1.4	1.3	1.2	1.1
Baseline	32	926	940	967	999	1040	1098
Optimized (GA)	104	114	132	163	428	506	621

Figures 3.8 – 3.21 show the change in the pressure coefficient c_p around the original Busemann biplane airfoil as the Mach number increases and decreases within the design-points range. The corresponding c_p contours for the optimized Busemann biplane airfoil are shown in Figures 3.22 – 3.35 to illustrate the wave-cancelling effect.

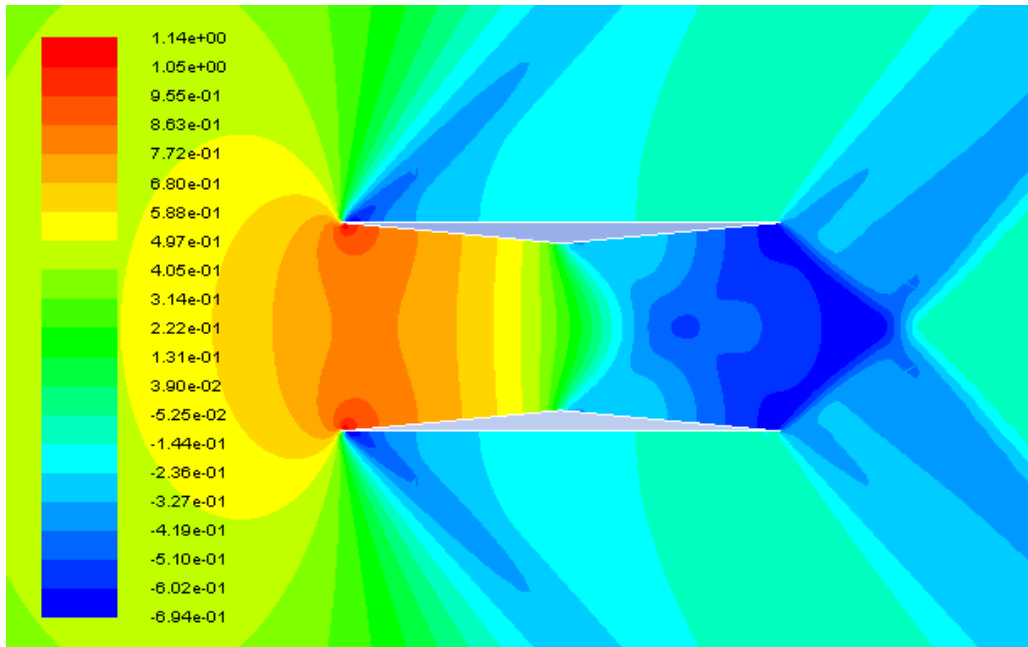


Figure 3.8 c_p contours around the original Busemann biplane airfoil at $M = 1.1$ during acceleration

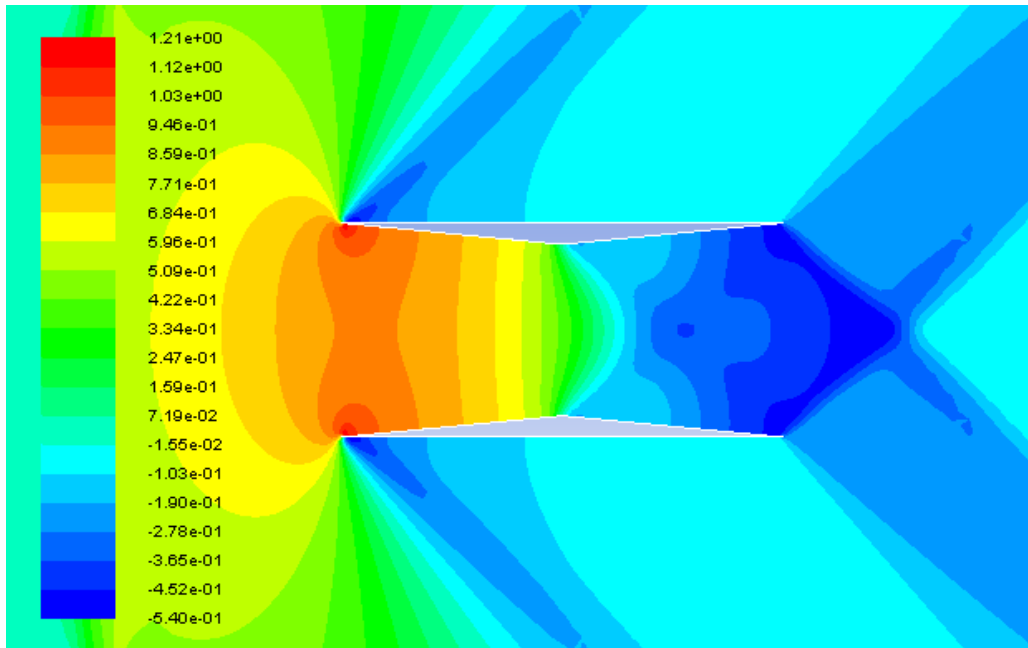


Figure 3.9 c_p contours around the original Busemann biplane airfoil at $M = 1.2$ during acceleration

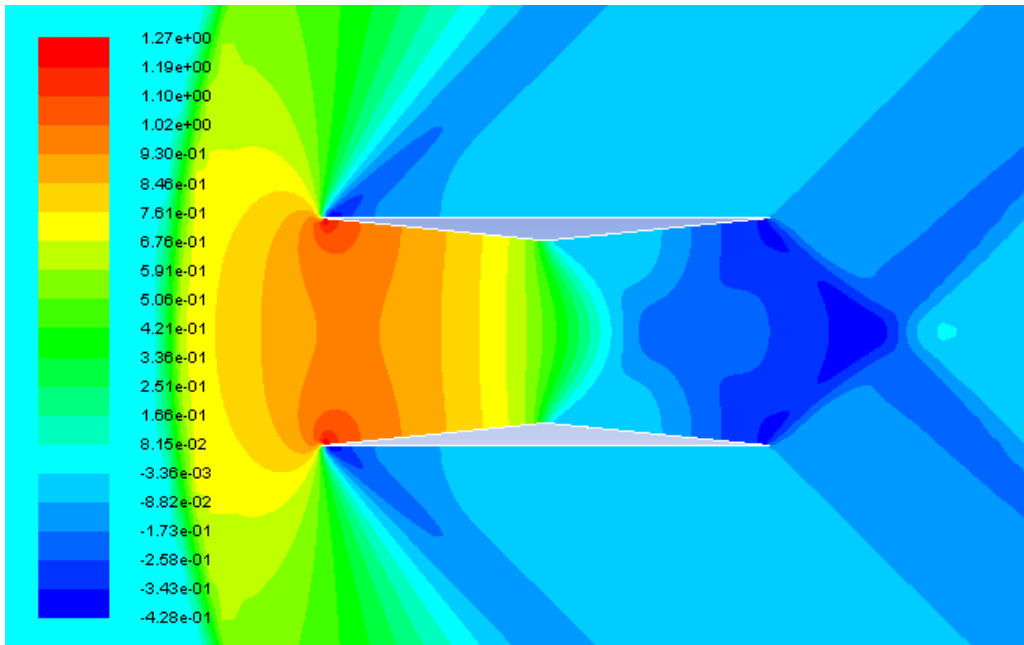


Figure 3.10 c_p contours around the original Busemann biplane airfoil at $M = 1.3$ during acceleration

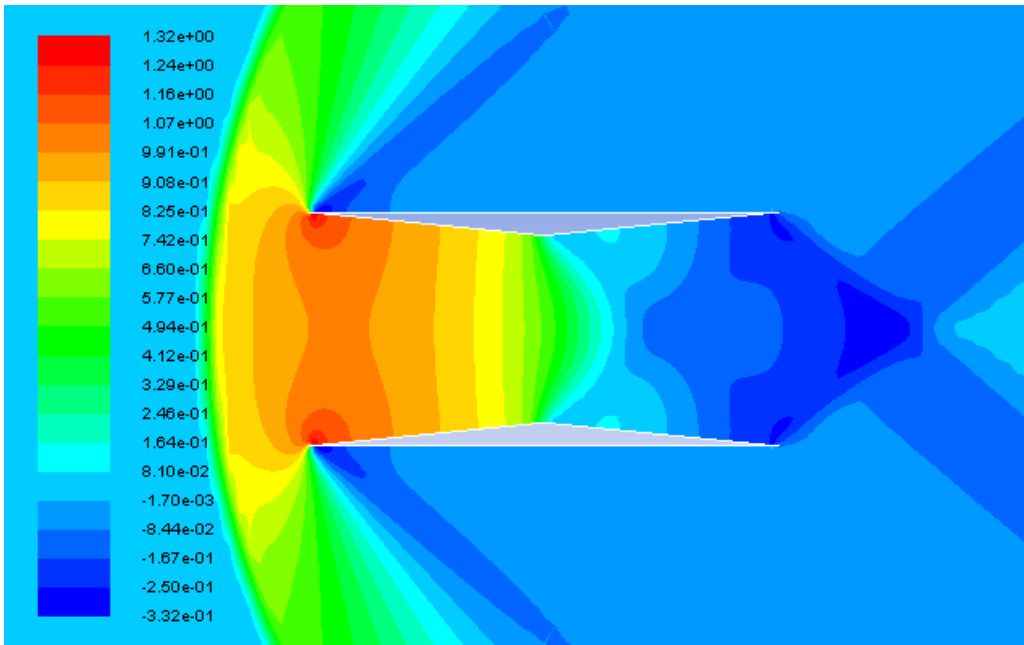


Figure 3.11 c_p contours around the original Busemann biplane airfoil at $M = 1.4$ during acceleration

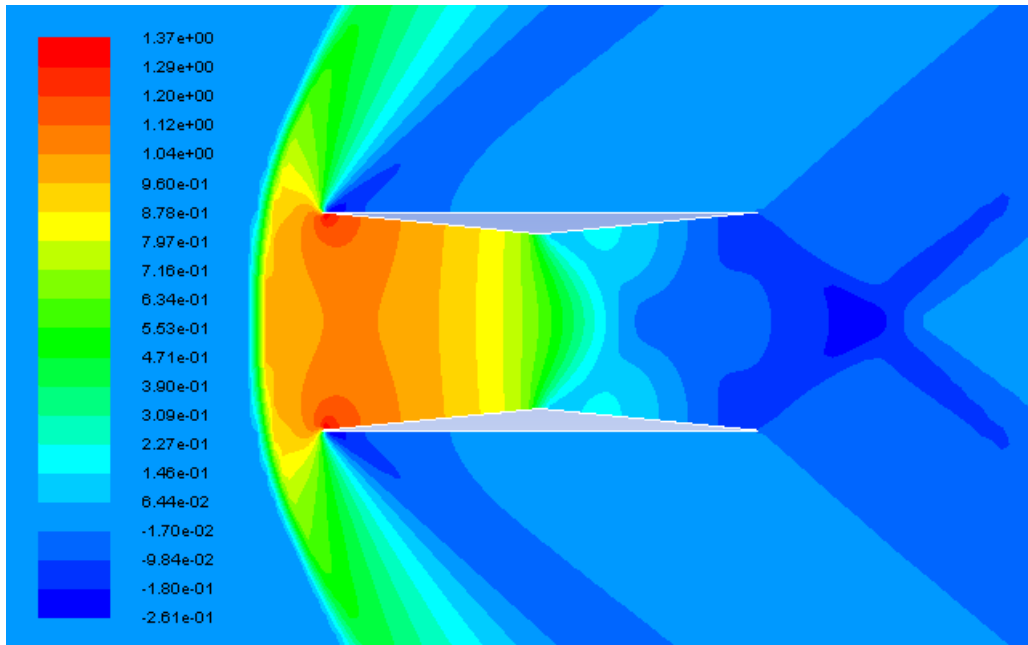


Figure 3.12 c_p contours around the original Busemann biplane airfoil at $M = 1.5$ during acceleration

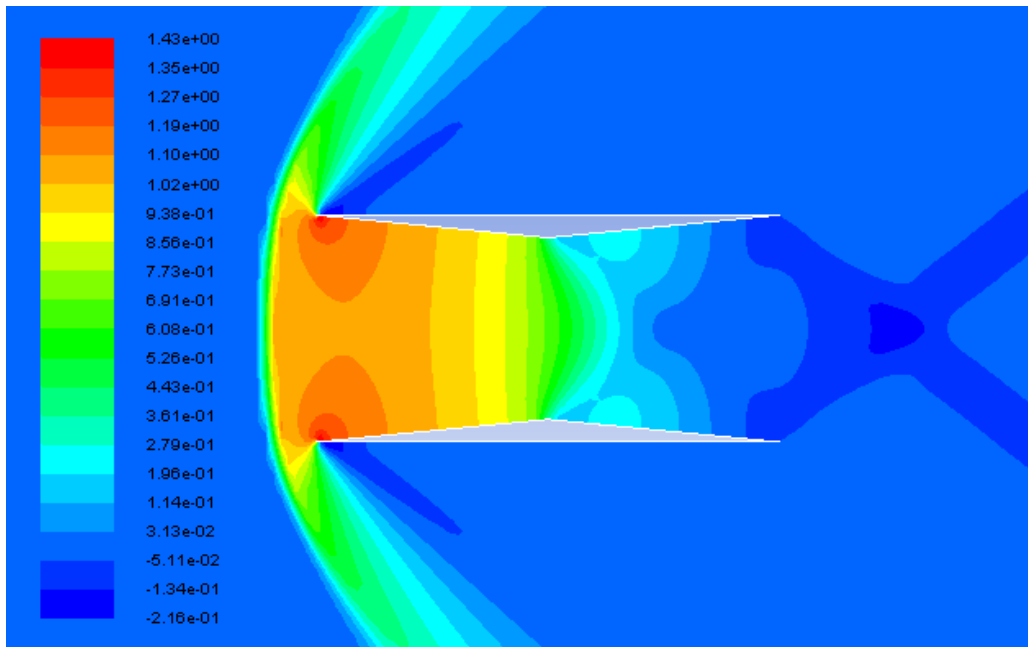


Figure 3.13 c_p contours around the original Busemann biplane airfoil at $M = 1.6$ during acceleration

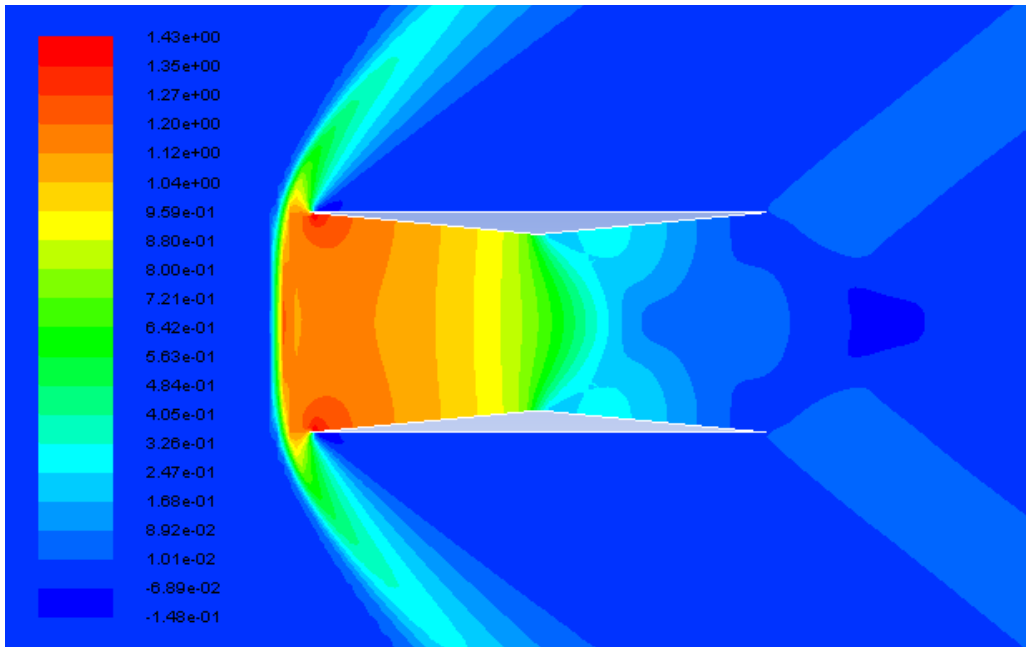


Figure 3.14 c_p contours around the original Busemann biplane airfoil at $M = 1.7$ during acceleration

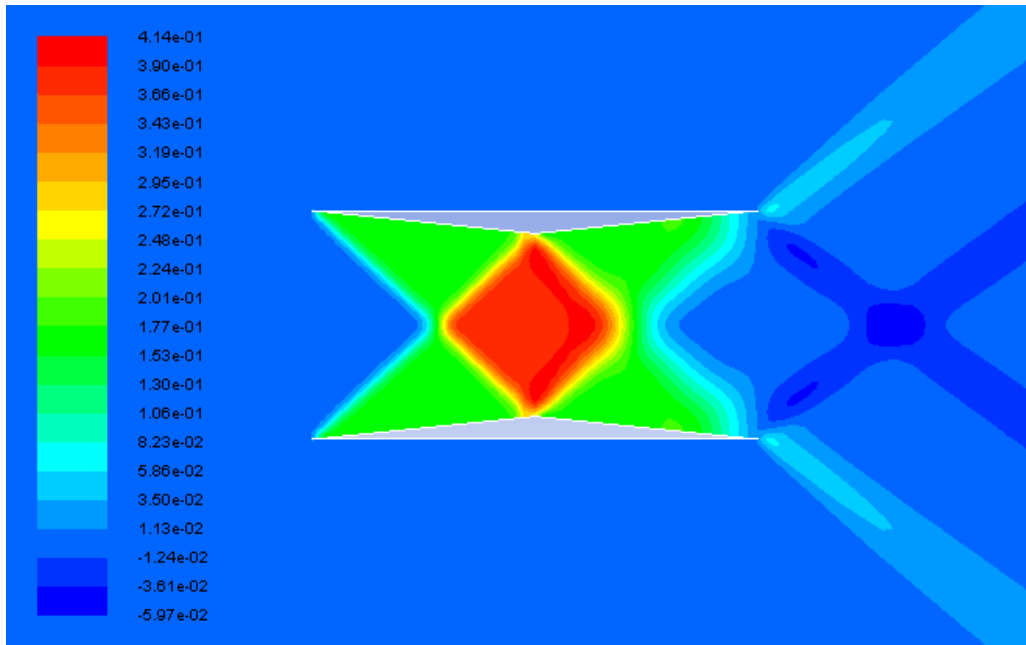


Figure 3.15 c_p contours around the original Busemann biplane airfoil at $M = 1.7$ during deceleration

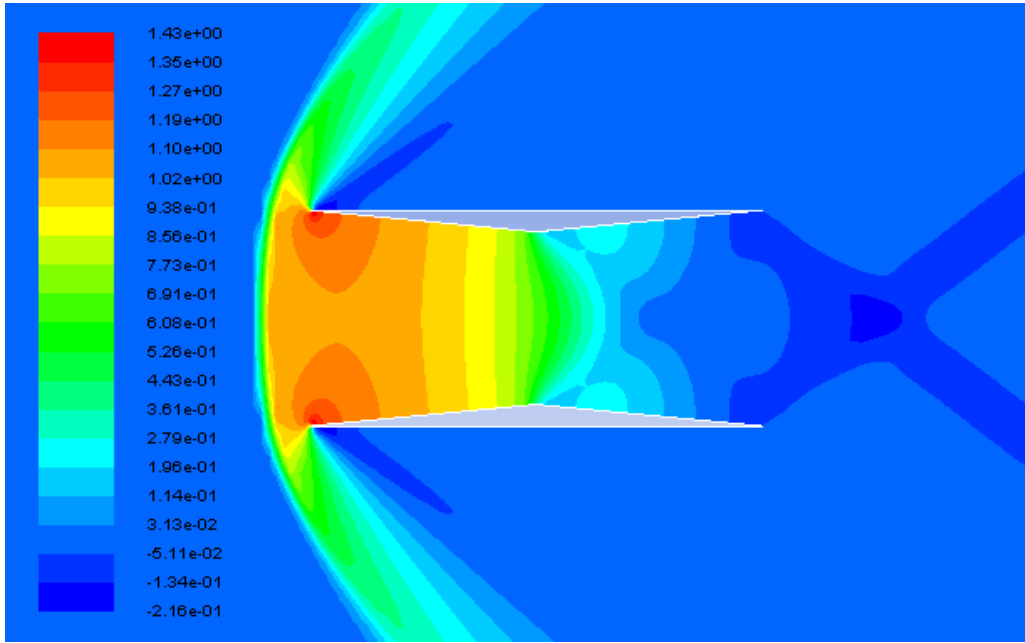


Figure 3.16 c_p contours around the original Busemann biplane airfoil at $M = 1.6$ during deceleration

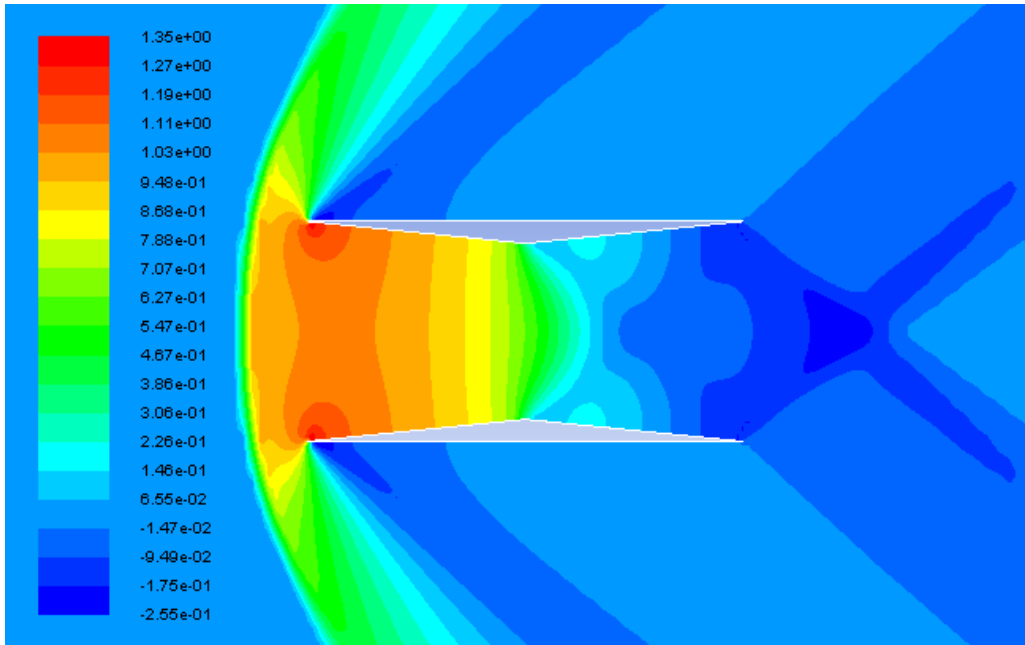


Figure 3.17 c_p contours around the original Busemann biplane airfoil at $M = 1.5$ during deceleration

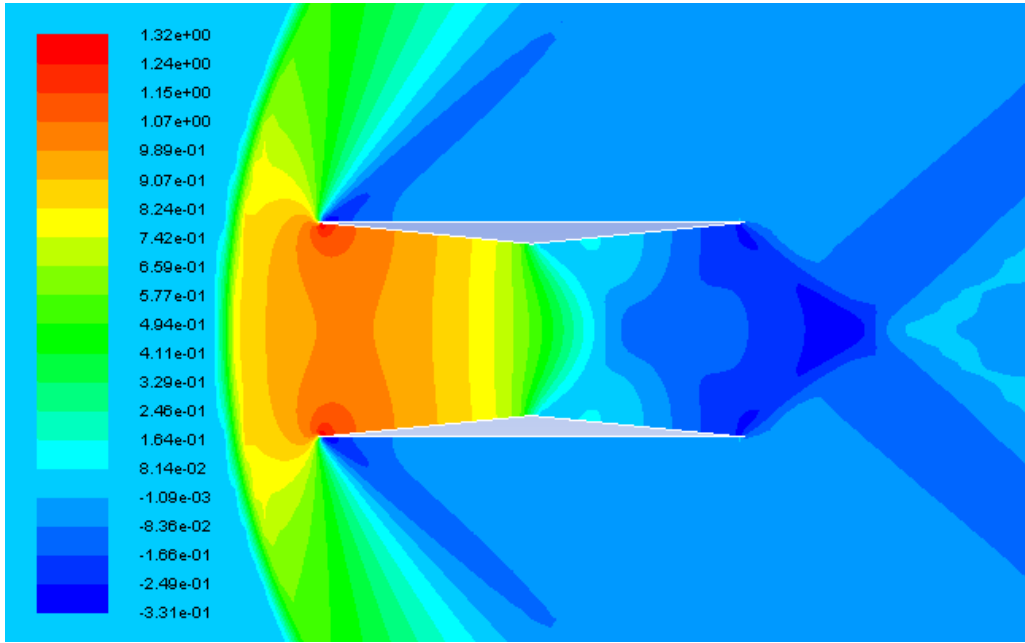


Figure 3.18 c_p contours around the original Busemann biplane airfoil at $M = 1.4$ during deceleration

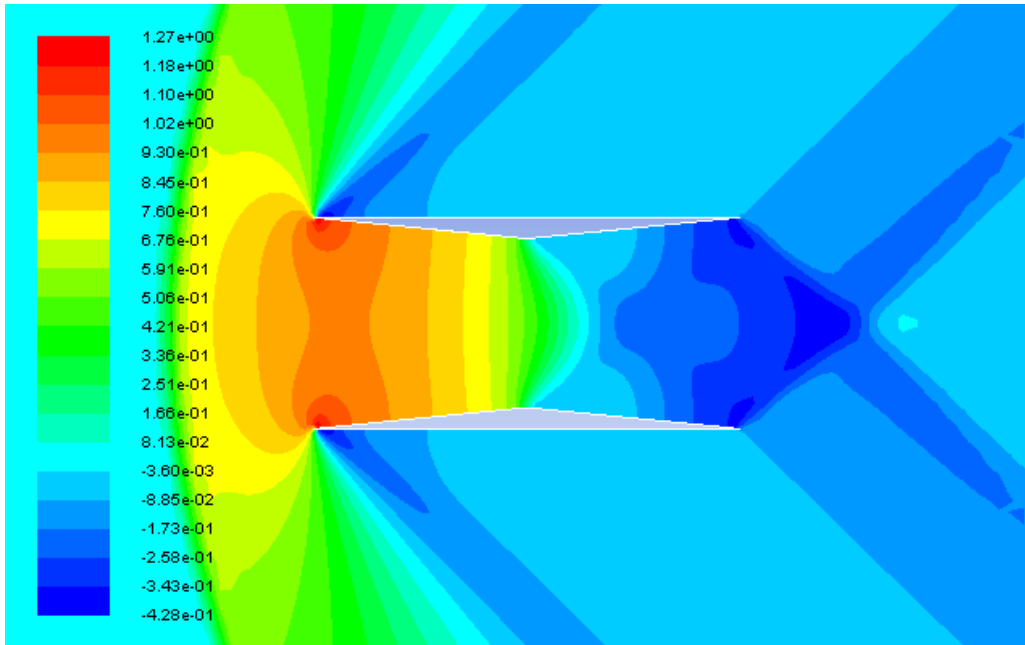


Figure 3.19 c_p contours around the original Busemann biplane airfoil at $M = 1.3$ during deceleration

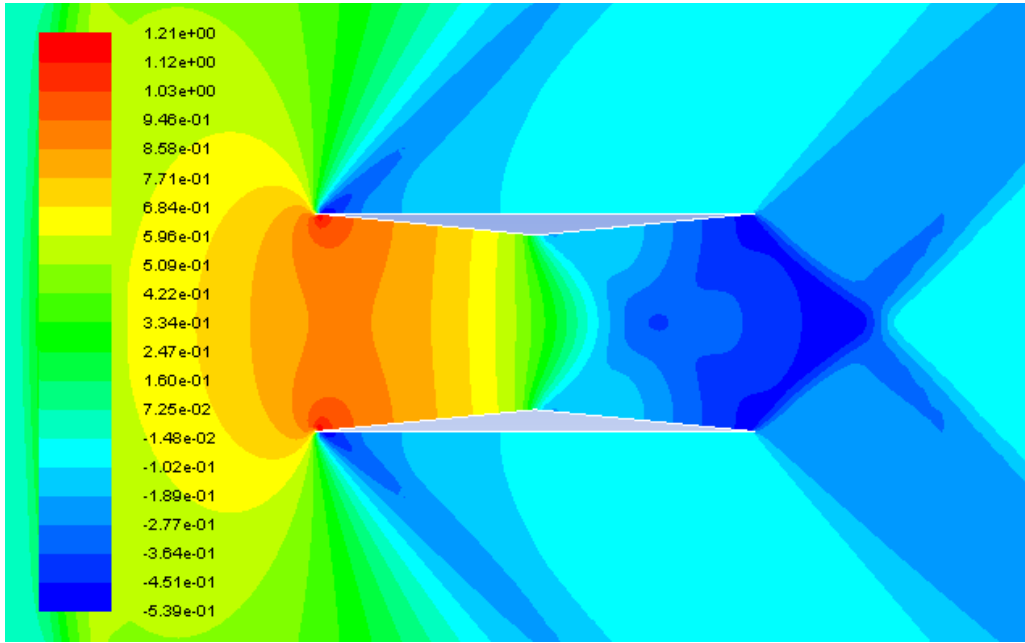


Figure 3.20 c_p contours around the original Busemann biplane airfoil at $M = 1.2$ during deceleration

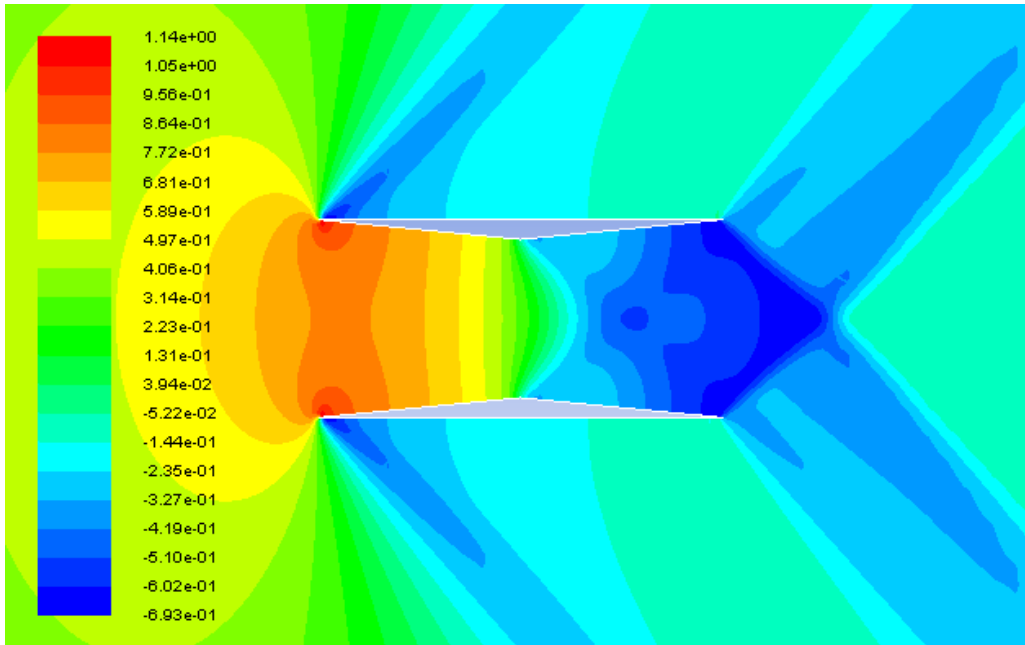


Figure 3.21 c_p contours around the original Busemann biplane airfoil at $M = 1.1$ during deceleration

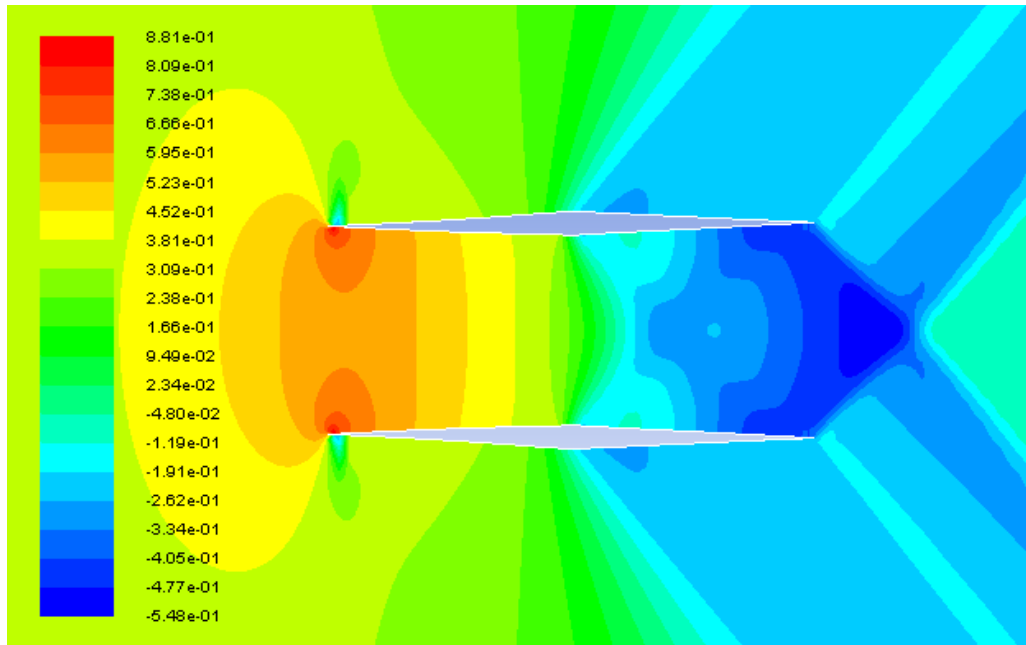


Figure 3.22 c_p contours around the optimized Busemann biplane airfoil at $M = 1.1$ during acceleration

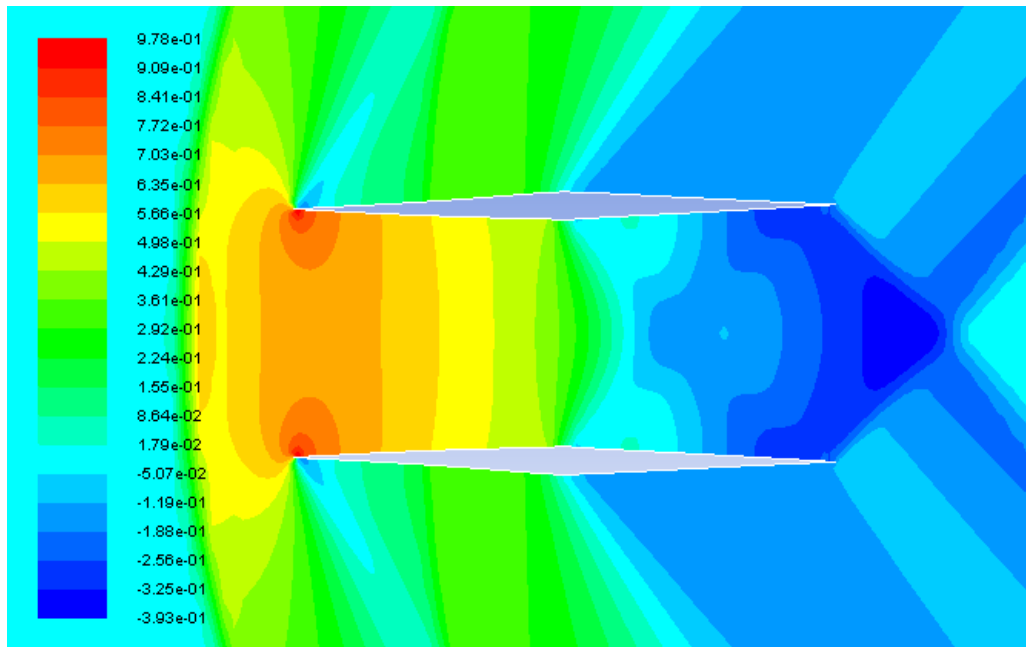


Figure 3.23 c_p contours around the optimized Busemann biplane airfoil at $M = 1.2$ during acceleration

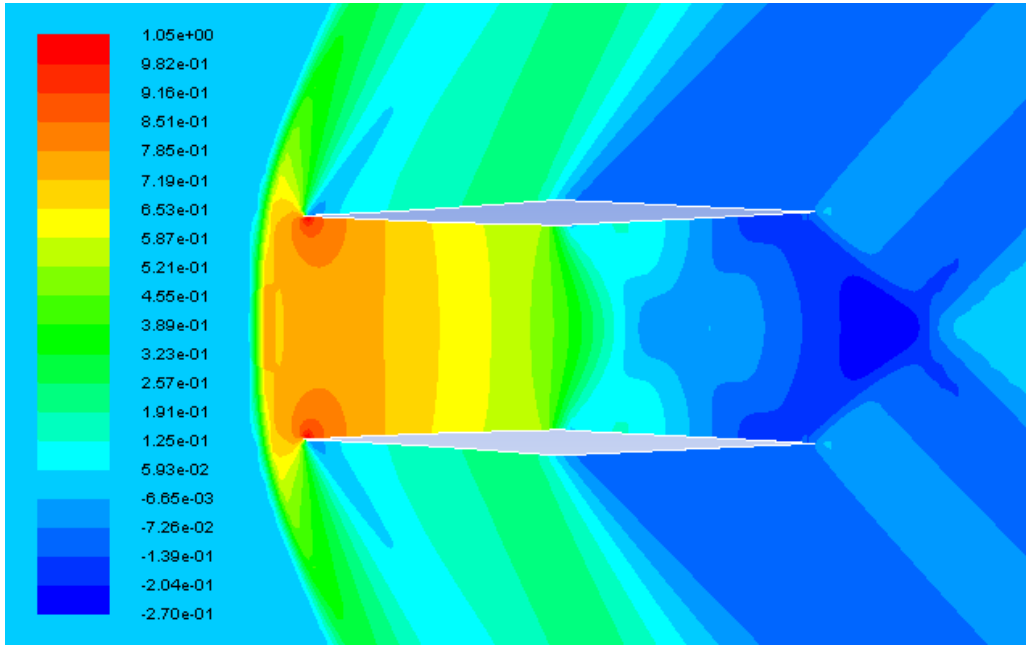


Figure 3.24 c_p contours around the optimized Busemann biplane airfoil at $M = 1.3$ during acceleration

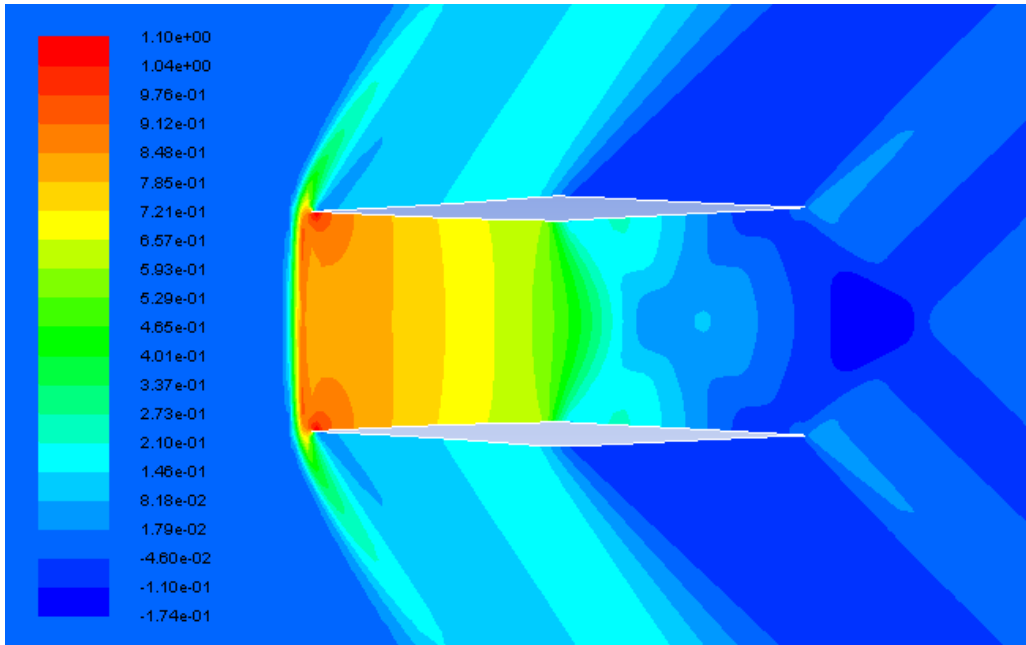


Figure 3.25 c_p contours around the optimized Busemann biplane airfoil at $M = 1.4$ during acceleration

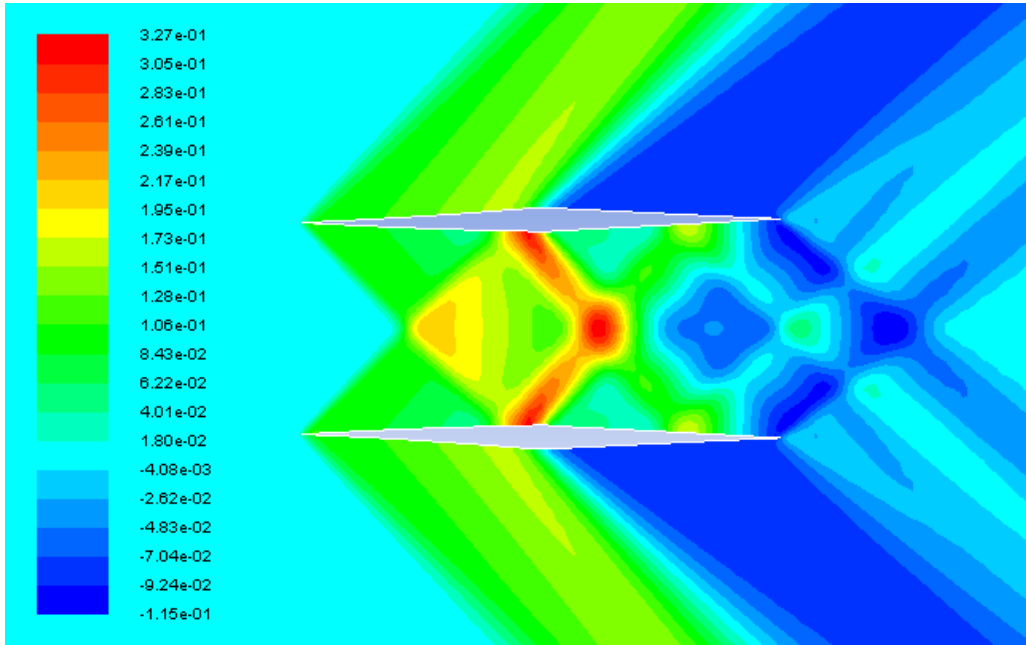


Figure 3.26 c_p contours around the optimized Busemann biplane airfoil at $M = 1.5$ during acceleration

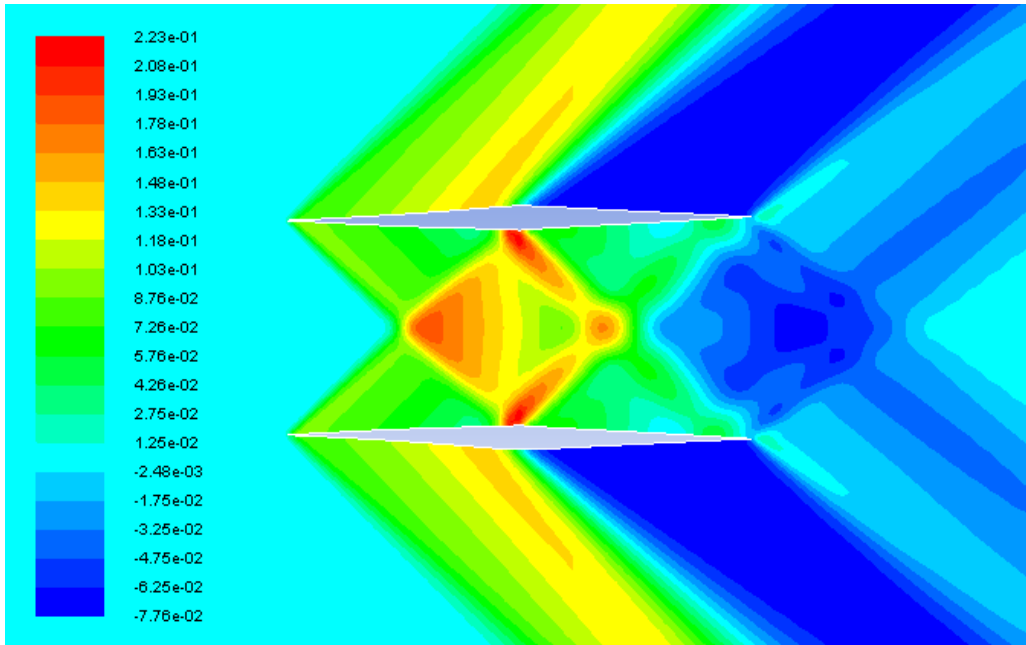


Figure 3.27 c_p contours around the optimized Busemann biplane airfoil at $M = 1.6$ during acceleration

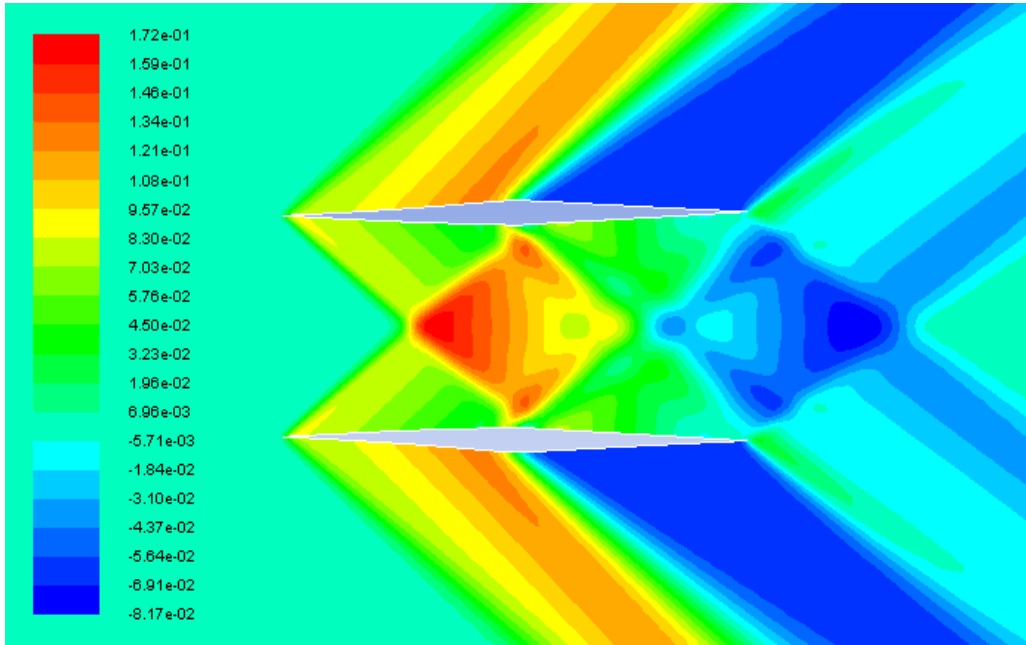


Figure 3.28 c_p contours around the optimized Busemann biplane airfoil at $M = 1.7$ during acceleration

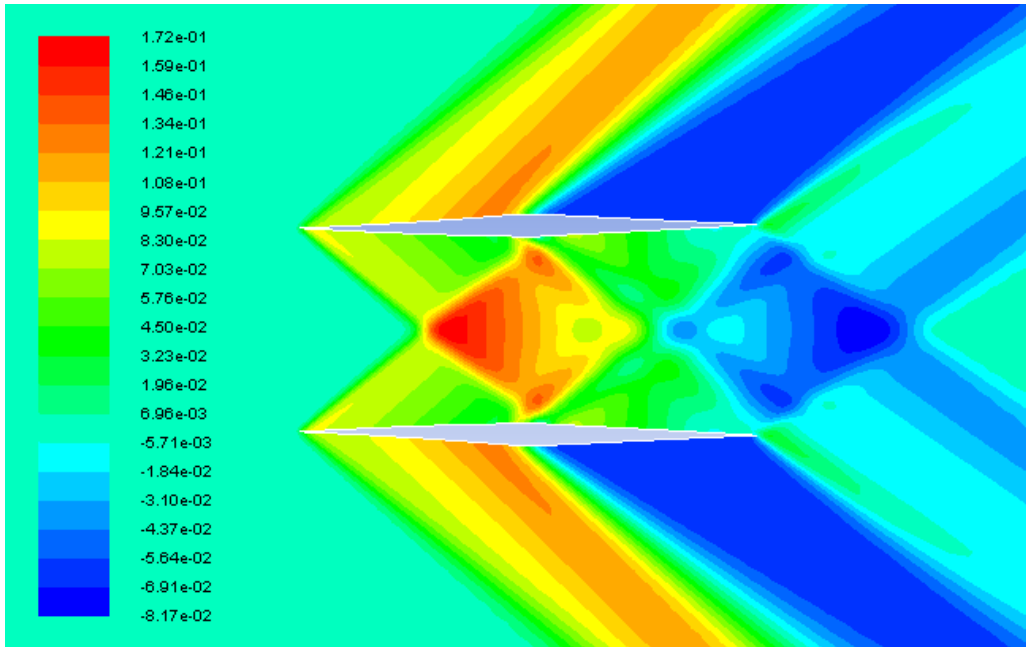


Figure 3.29 c_p contours around the optimized Busemann biplane airfoil at $M = 1.7$ during deceleration

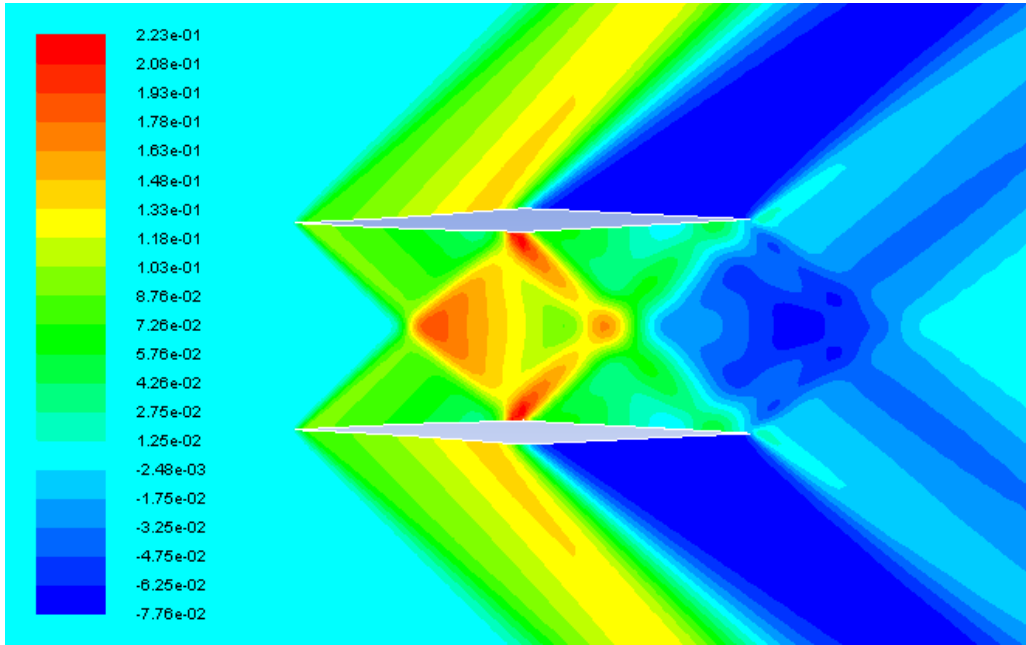


Figure 3.30 c_p contours around the optimized Busemann biplane airfoil at $M = 1.6$ during deceleration

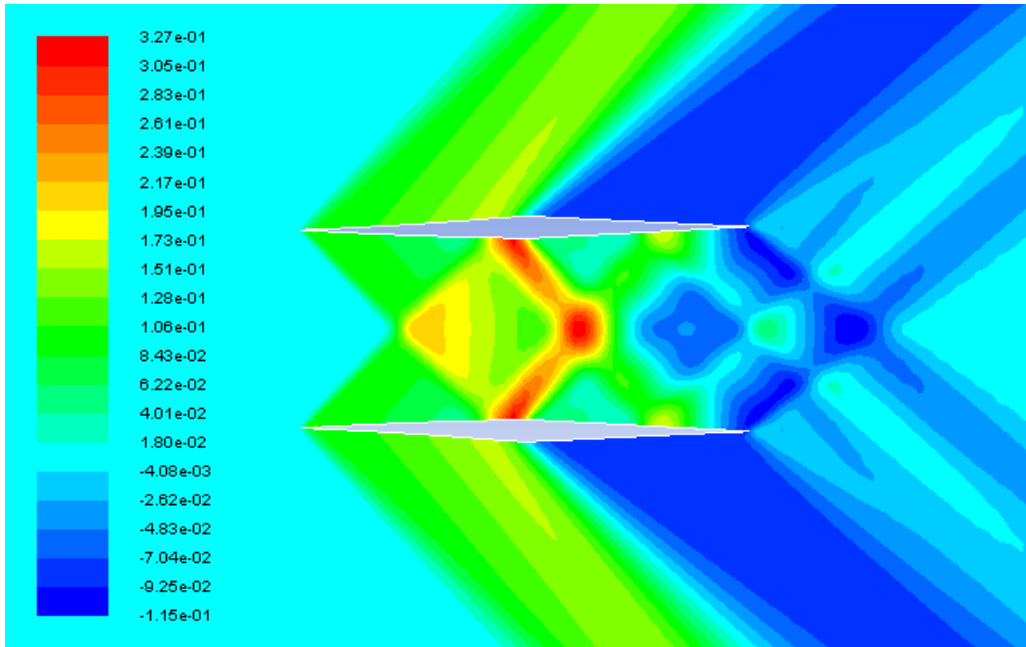


Figure 3.31 c_p contours around the optimized Busemann biplane airfoil at $M = 1.5$ during deceleration

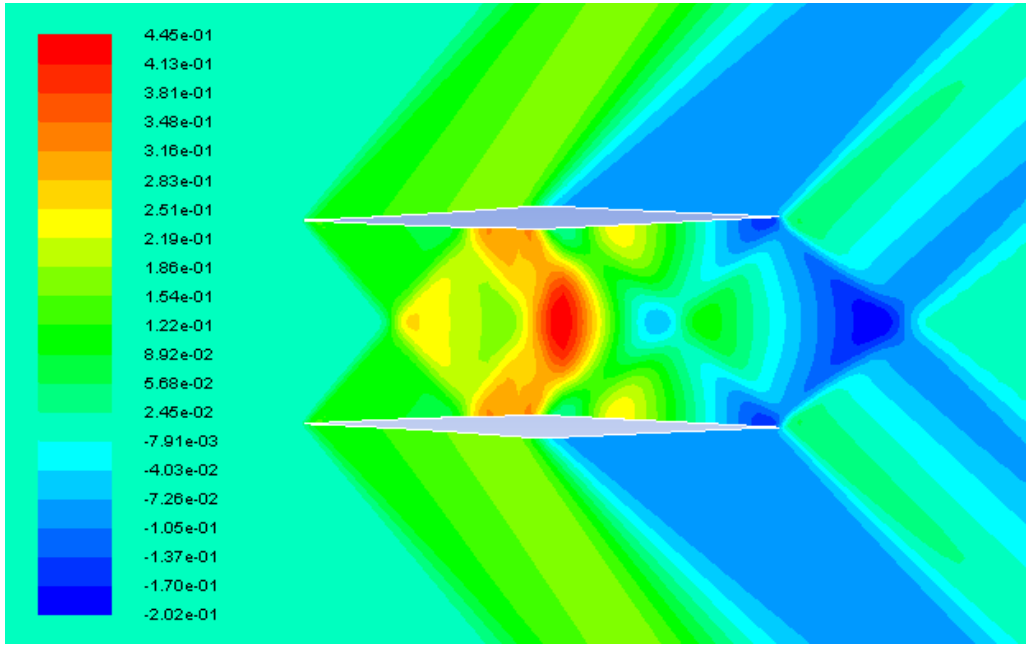


Figure 3.32 c_p contours around the optimized Busemann biplane airfoil at $M = 1.4$ during deceleration

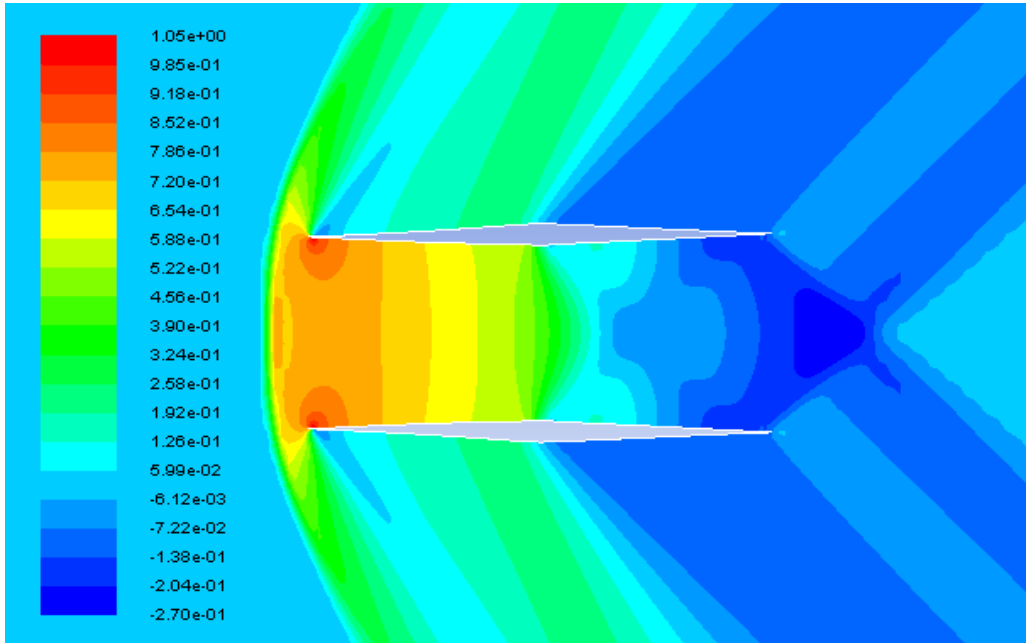


Figure 3.33 c_p contours around the optimized Busemann biplane airfoil at $M = 1.3$ during deceleration

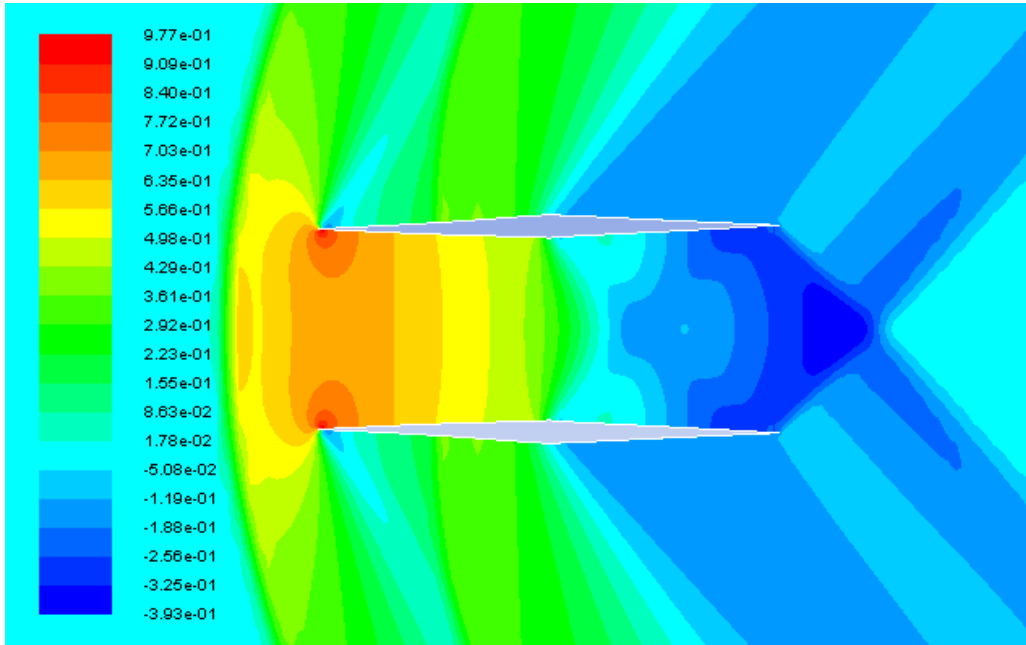


Figure 3.34 c_p contours around the optimized Busemann biplane airfoil at $M = 1.2$ during deceleration

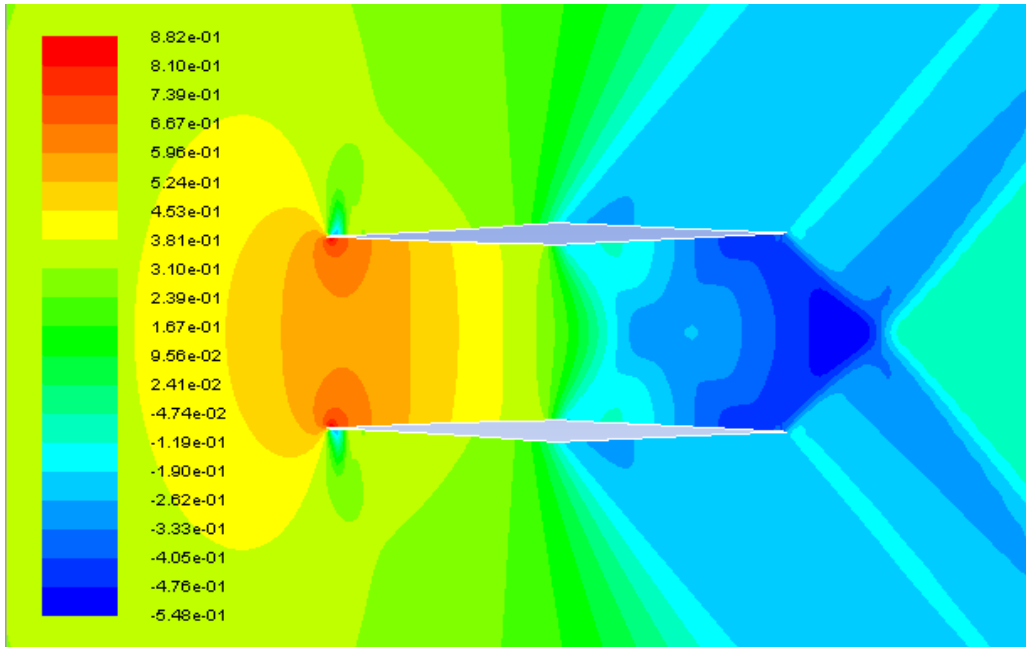


Figure 3.35 c_p contours around the optimized Busemann biplane airfoil at $M = 1.1$ during deceleration

Figure 3.36 shows the comparison of the drag coefficients for the standard diamond-shaped airfoil, the baseline Busemann biplane airfoil and the optimized Busemann biplane airfoil under nonlifting condition. As shown in the figure, the separation between the acceleration and deceleration lines for c_d still exists for the optimized Busemann biplane airfoil, which means that the flow-hysteresis and the choked-flow effects are not totally eliminated. However, as clearly shown in Figure 3.36, the flow-hysteresis area has been significantly reduced and the drag increase during deceleration due to the choked-flow phenomenon is much smaller than that for the original Busemann biplane airfoil. The drag of the optimized Busemann biplane airfoil in the subsonic region is also smaller than that of the original Busemann biplane airfoil, although it is slightly higher than that of the standard diamond-shaped airfoil for $0.6 < M < 0.85$. For both subsonic and supersonic conditions, the optimized Busemann biplane airfoil has been able to significantly reduce the wave drag compared to the original Busemann biplane airfoil. At the design condition Mach number of 1.7 however, the drag coefficient of the optimized Busemann biplane airfoil ($c_d = 0.01038$) is much higher than that of the original Busemann biplane airfoil ($c_d = 0.002182$). This is due to the fact that for the shape optimization, our focus has been reducing the flow-hysteresis and choked-flow effects and we chose to assign equal weights to all Mach numbers used as the multiple design points. To address this problem, we could have put more weight on the design condition ($M = 1.7$) during the optimization process.

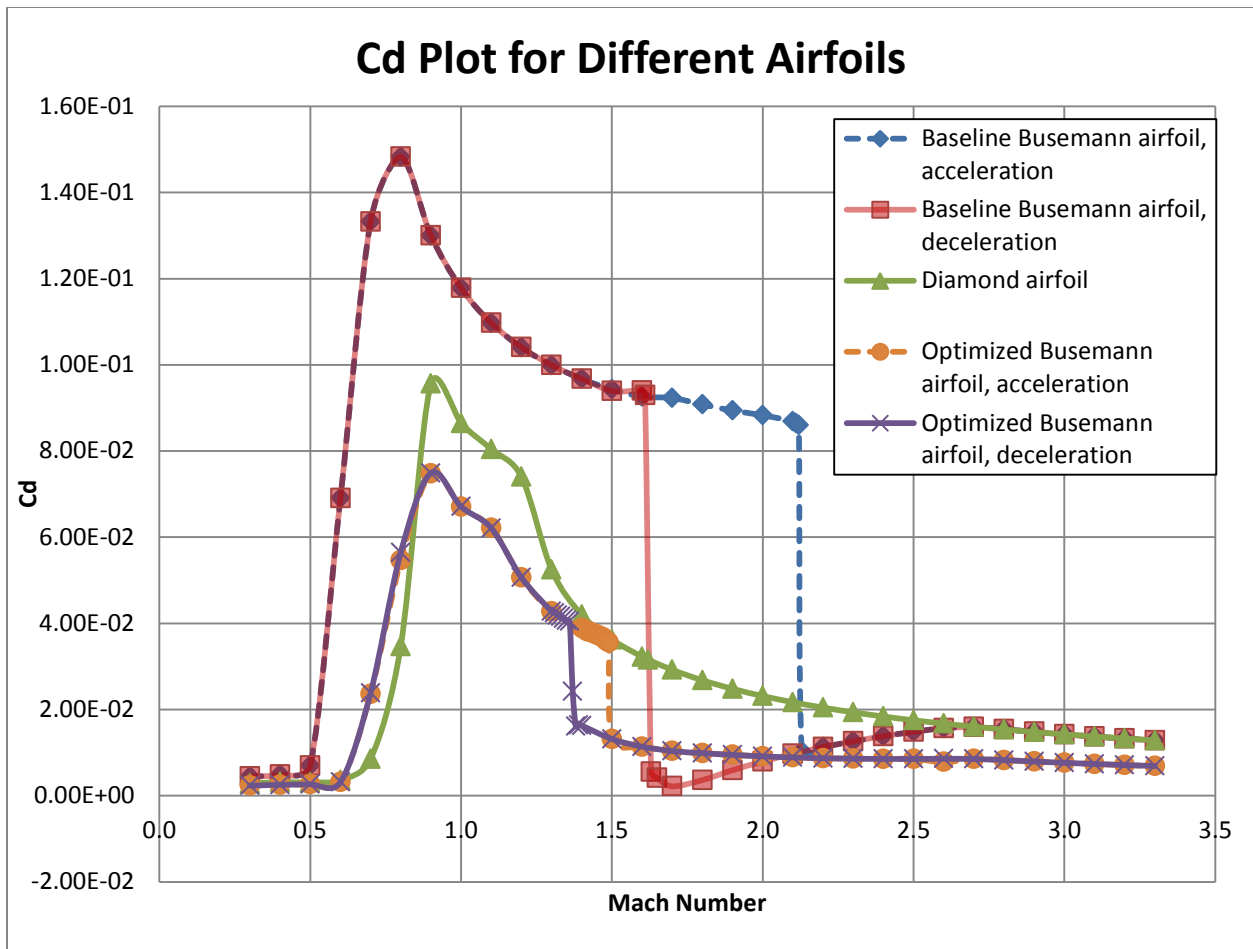


Figure 3.36 c_d plot for different airfoils under nonlifting condition

Next we examine the details of the flow field during acceleration and deceleration for the optimized Busemann biplane airfoil and compare them with the original baseline Busemann biplane airfoil and the optimization results obtained by Hu et al. [17] using an adjoint-based optimization technique. Figure 3.37 and Figure 3.38 show the pressure coefficient contours of the optimized Busemann biplane airfoil under acceleration and deceleration respectively. Figure 3.39 and Figure 3.40 show the pressure coefficient contours of the optimized Busemann biplane airfoil using the adjoint-based technique [17] under acceleration and deceleration respectively. During acceleration, the flow-hysteresis effect still exists and a bow shock wave is formed in front of the airfoil. The swallowing of the bow shock wave happens when the Mach number increases from 1.49 to 1.50 in our GA optimization as shown in Figure 3.37(f); it happens when Mach number increases from 1.52 to 1.53 in the adjoint-based optimization as shown in Figure 3.39(e) and it happens when Mach number

increases from 2.12 to 2.13 for the original Busemann biplane airfoil as shown in Figure 2.16(f). The drag coefficient decreases from 0.03556 to 0.01316 in the present GA based optimization and from 0.03336 to 0.01221 in the adjoint-based optimization [17].

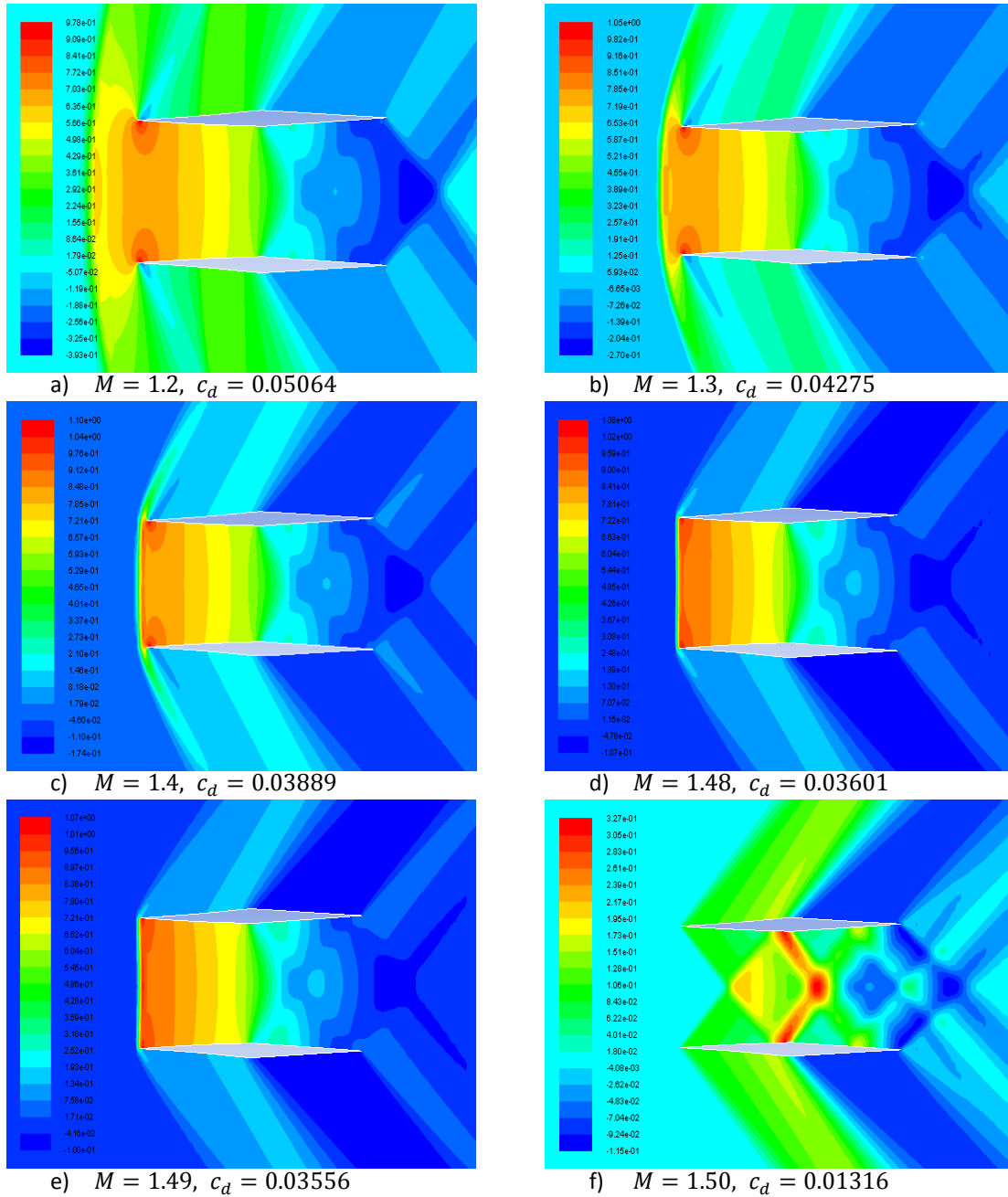


Figure 3.37 c_p -contours of the GA-optimized Busemann biplane airfoil with zero-lift during acceleration

During deceleration, the choked-flow effect still exists, however, it is shifted to a lower Mach number of 1.36 in the GA optimization and 1.37 in the adjoint-based optimization; it occurs at Mach number of 1.6 in the original baseline Busemann airfoil.

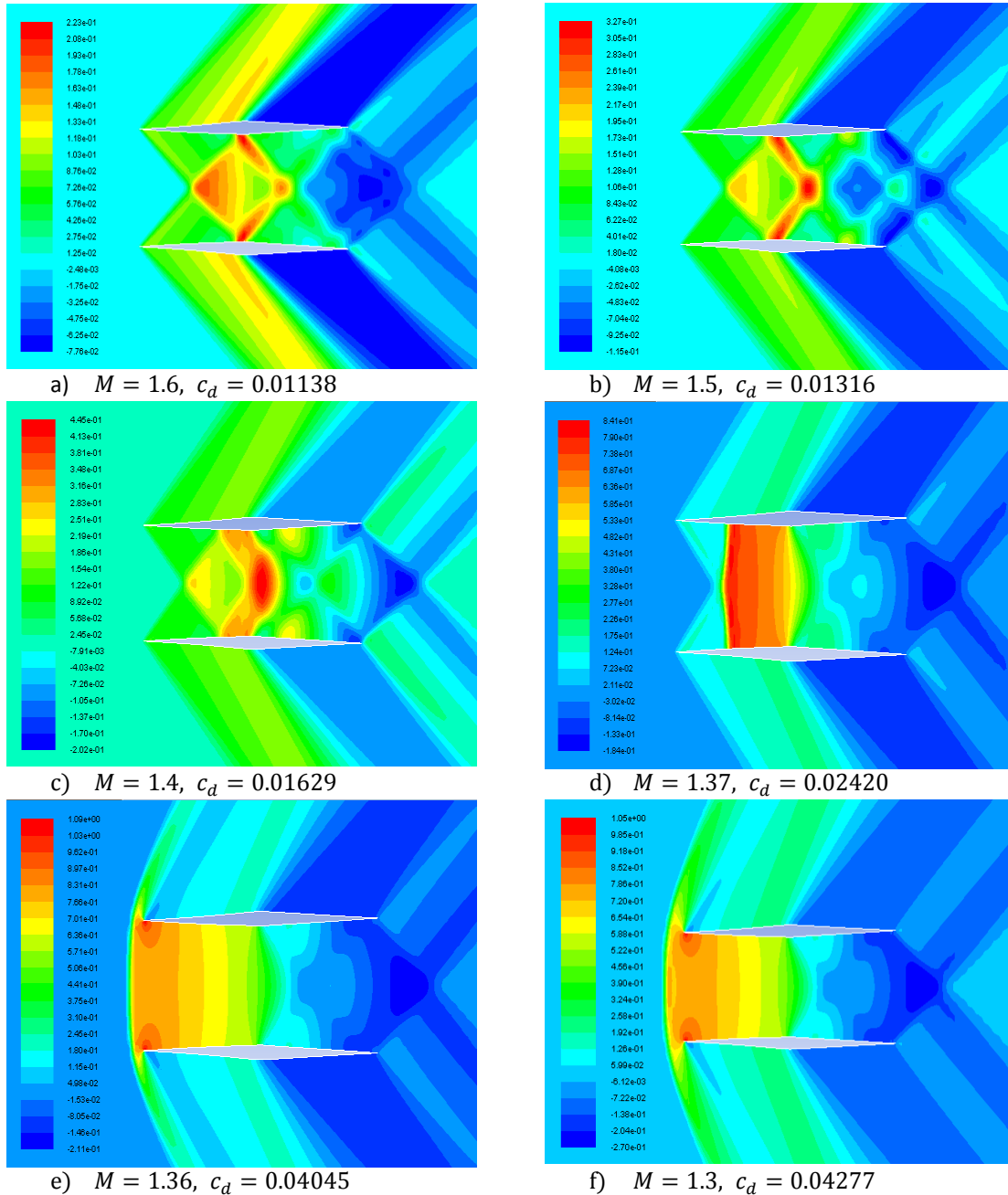


Figure 3.38 c_p -contours of the GA-optimized Busemann biplane airfoil with zero-lift during deceleration

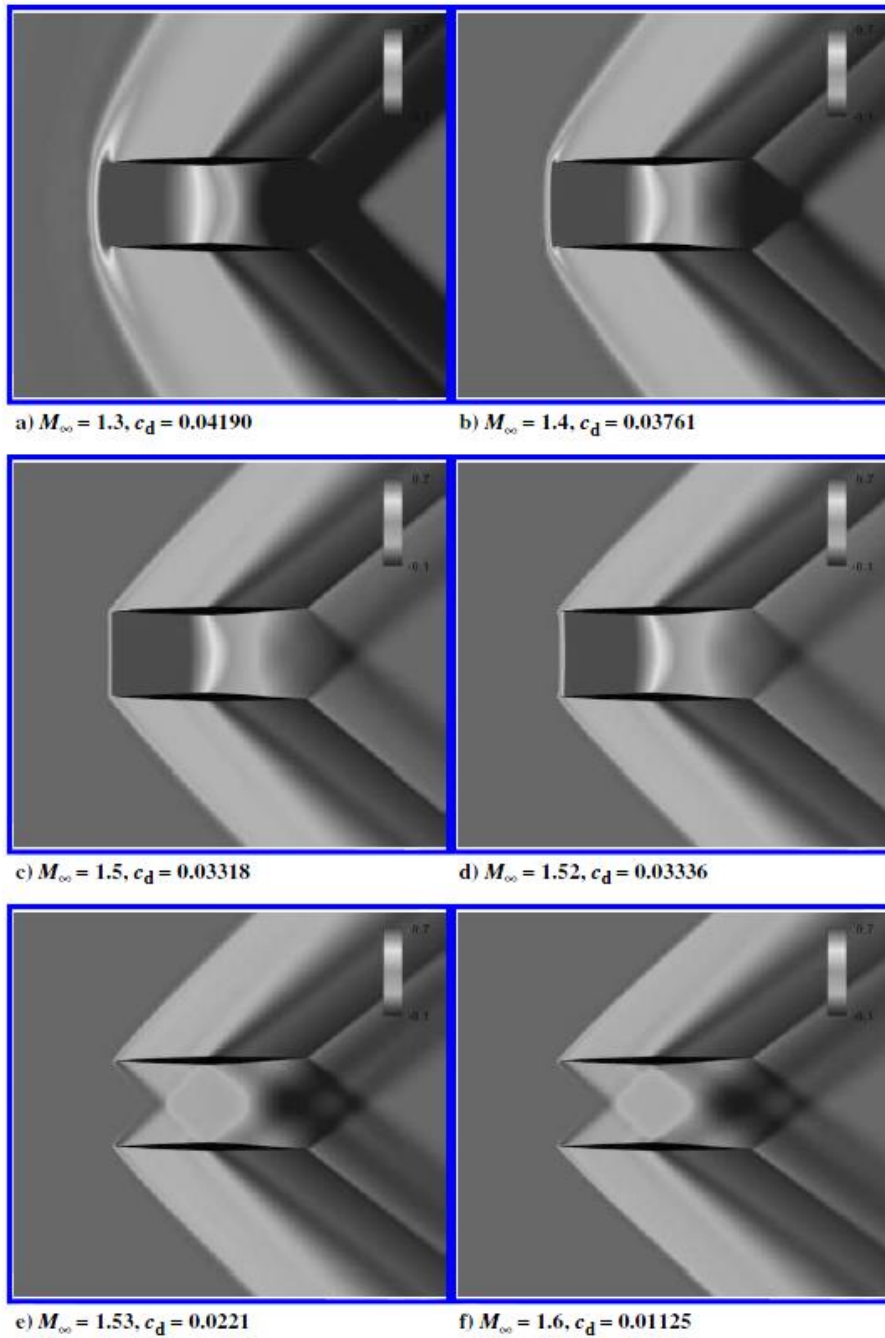


Figure 3.39 c_p -contours of the Adjoint-based-optimized Busemann biplane airfoil with zero-lift during acceleration [17]

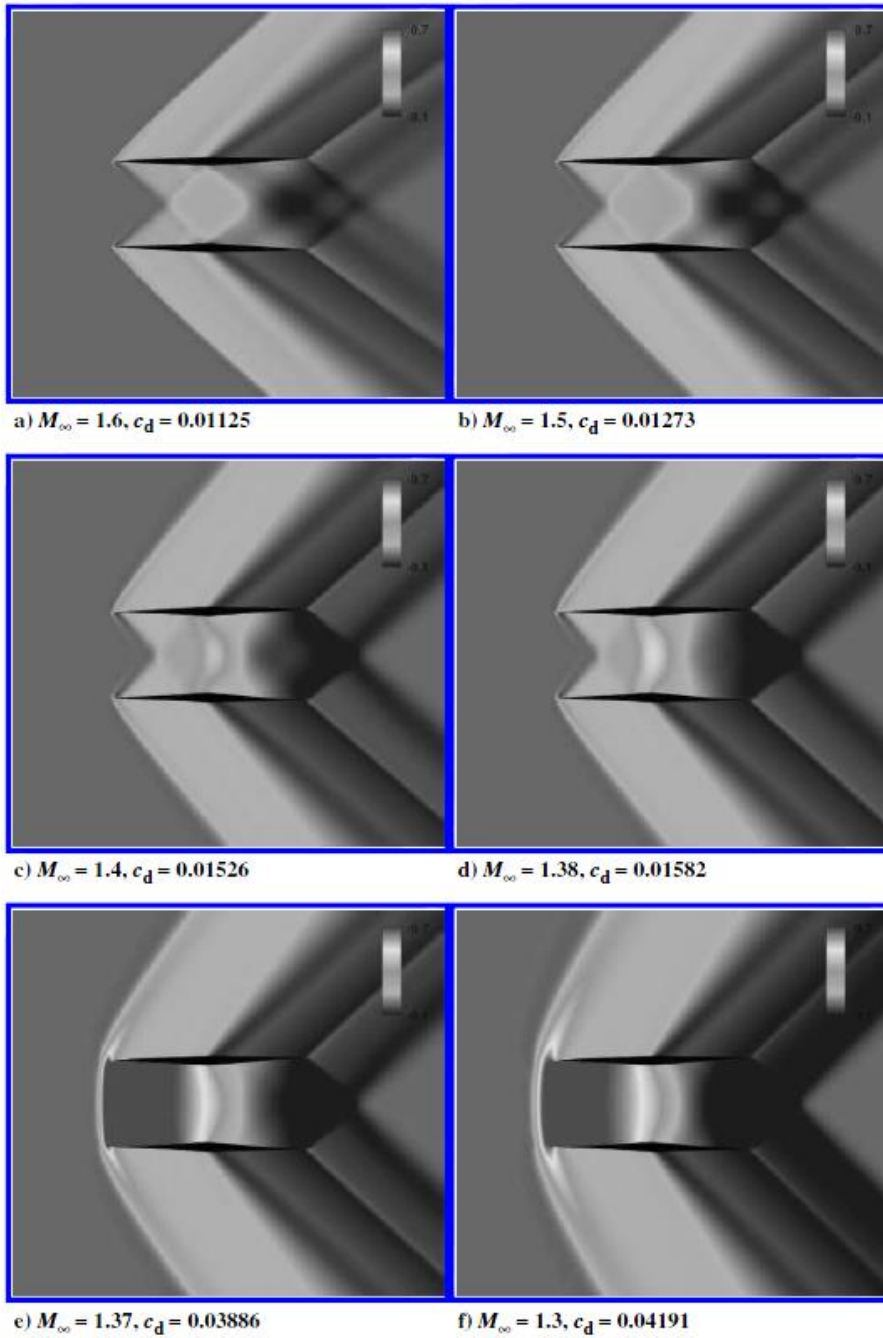


Figure 3.40 c_p -contours of the Adjoint-based-optimized Busemann biplane airfoil with zero-lift during deceleration [17]

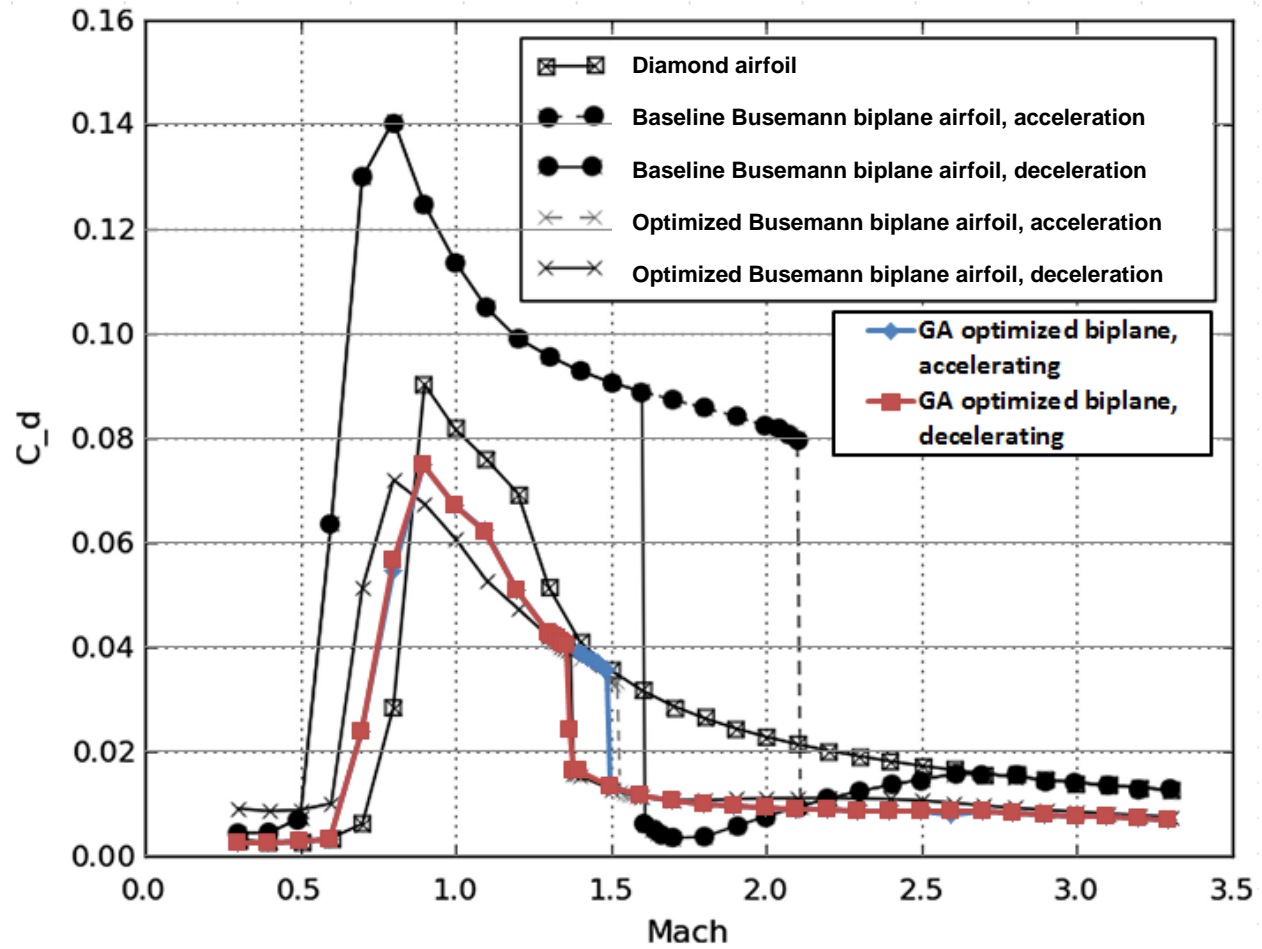


Figure 3.41 c_d plot for different airfoils under nonliting condition

In conclusion, as shown in Figure 3.41, the drag coefficient of the GA-optimized Busemann biplane airfoil is significantly reduced comparing to the original Buseman biplane airfoil and it matches with the adjoint-based optimization result obtained by Hu et al. [17].

3.3 Shape Optimization of Busemann Biplane Airfoil under Lifting Condition

3.3.1 Application of Multi-objective Genetic Algorithm (MOGA)

For the lifting condition with the lift coefficient $c_l \neq 0$, a Multi-Objective Genetic Algorithm (MOGA) is employed for shape optimization of the Busemann biplane airfoil. The two objectives to be achieved are to minimize the drag coefficient c_d while maximizing the lift coefficient c_l . Similar to that for the nonlifting condition described in section 3.2, a total of seven design points ranging from $M = 1.1$ to $M = 1.7$ are used during the optimization process. For the fitness functions for the lifting case, we use the sum of evenly-weighted average of both c_d and c_l . The GA parameters used for the lifting case are listed in Table 3.7.

Table 3.7 GA parameters for shape optimization of Busemann biplane airfoil under lifting condition

GA Parameters	Description
Generation Size	8 individuals per generation
Number of Generations	Maximum of 50 generations if convergence not obtained
Number of Design Variables	28 in total, 14 (7 for c_d & 7 for c_l) for acceleration and 14 (7 for c_d & 7 for c_l) for deceleration
Selection Type	Roulette Wheel Selection
Crossover Rate	0.7
Mutation Rate	0.1
Error of Mutation Constant	0.8, which determines how much mutation affects the curves as generations go on

3.3.2 Airfoil Parameterization

Similar to that for the nonlifting case, the random shapes of the airfoil are generated by using the Bezier curves with control points. For the lifting case with $c_l \neq 0$, since the upper and lower components of the biplane airfoil are not symmetric, both the upper and lower component need to be defined separately. The thickness distribution for both the upper and lower components is still kept the same as for the nonlifting case. Now, a total of four Bezier curves are needed to define the shape of the airfoil. The range of the x- and y-coordinates of the Bezier curves for the two components of the airfoil are listed in Table 3.8 and Table 3.9.

Table 3.8 Coordinate range for each control point of the Bezier curves for the upper component of the airfoil under lifting condition

	Coordinate	Upper Limit	Lower Limit
Bezier curve #1	x1	Fixed at x=-0.5	
	x2	-0.05	-0.5
	x3	Fixed at x=-0.05	
	x4	Fixed at x=0	
	y1	0.25	0.2
	y2	0.25	0.2
	y3	Fixed at y=0.25	
	y4	Fixed at y=0.25	
Bezier curve #2	x1	Fixed at x=0	
	x2	Fixed at x=0.05	
	x3	0.5	0.05
	x4	Fixed at x=0.5	
	y1	Fixed at y=0.25	
	y2	Fixed at y=0.25	
	y3	0.25	0.2
	y4	0.25	0.2

Table 3.9 Coordinate range for each control point of the Bezier curves for the lower component of the airfoil under lifting condition

	Coordinate	Upper Limit	Lower Limit
Bezier curve #3	x1	Fixed at x=-0.5	
	x2	-0.05	-0.5
	x3	Fixed at x=-0.05	
	x4	Fixed at x=0	
	y1	-0.2	-0.25
	y2	-0.2	-0.25
	y3	Fixed at y=-0.25	
	y4	Fixed at y=-0.25	
Bezier curve #4	x1	Fixed at x=0	
	x2	Fixed at x=0.05	
	x3	0.5	0.05
	x4	Fixed at x=0.5	
	y1	Fixed at y=-0.25	
	y2	Fixed at y=-0.25	
	y3	-0.2	-0.25
	y4	-0.2	-0.25

Figure 3.42 shows a randomly generated Busemann-type biplane airfoil shape using Bezier curves for the lifting condition. Figures 3.43 - 3.46 respectively show the detailed geometry of Bezier curves #1 to #4 along with their control points.

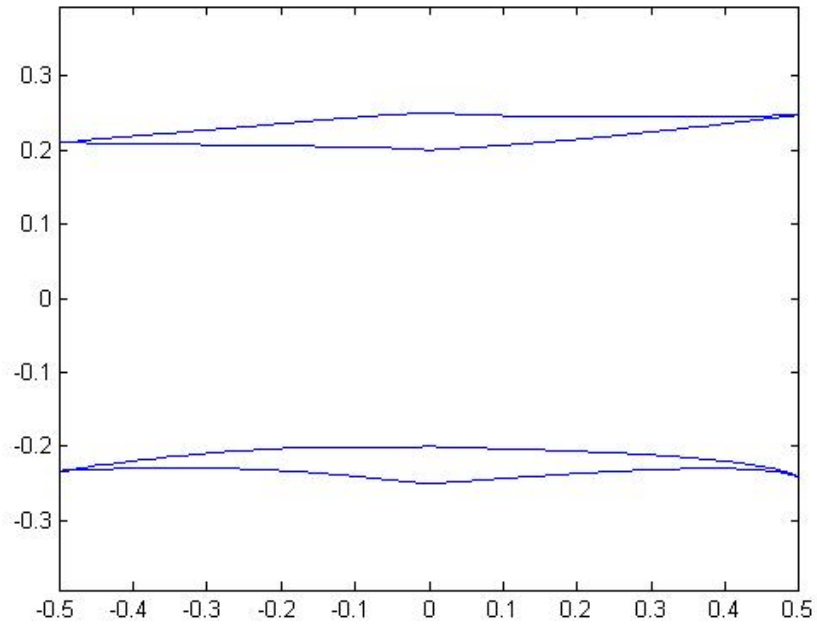


Figure 3.42 Randomly generated Busemann-type biplane airfoil shape using Bezier curves for lifting condition

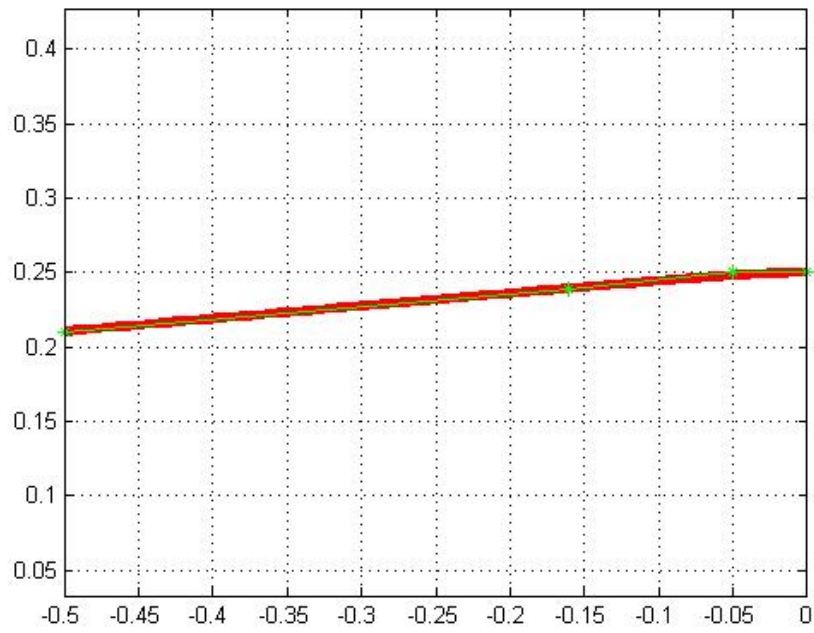


Figure 3.43 Detailed geometry of Bezier curve #1 and its control points

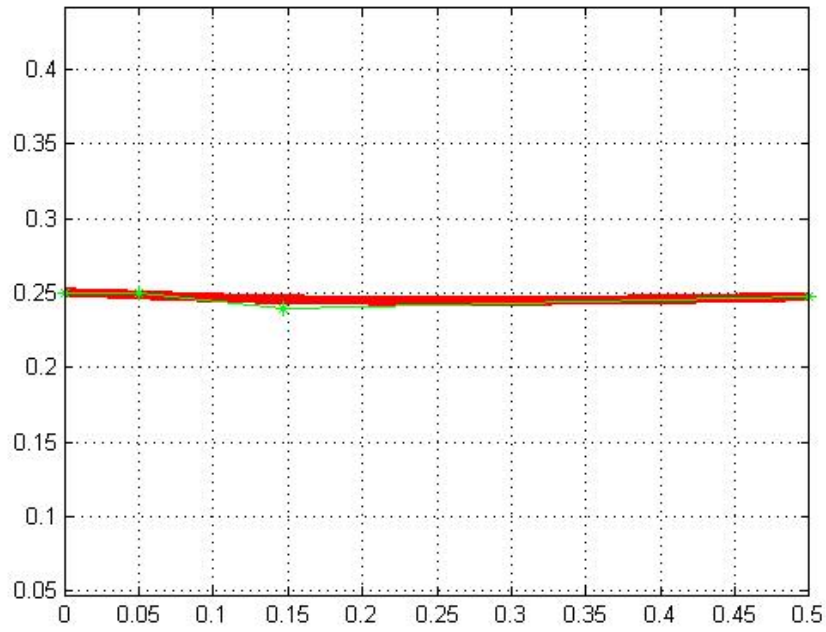


Figure 3.44 Detailed geometry of Bezier curve #2 and its control points

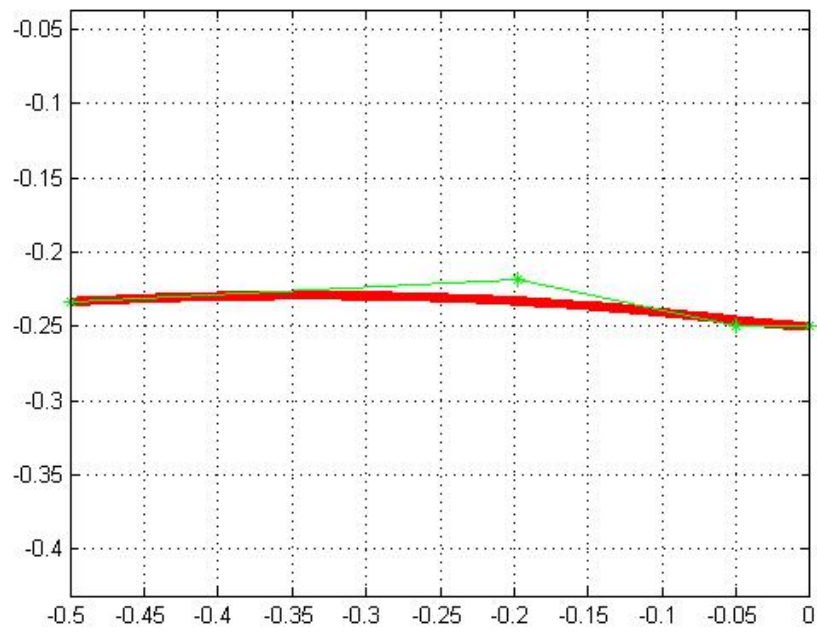


Figure 3.45 Detailed geometry of Bezier curve #3 and its control points

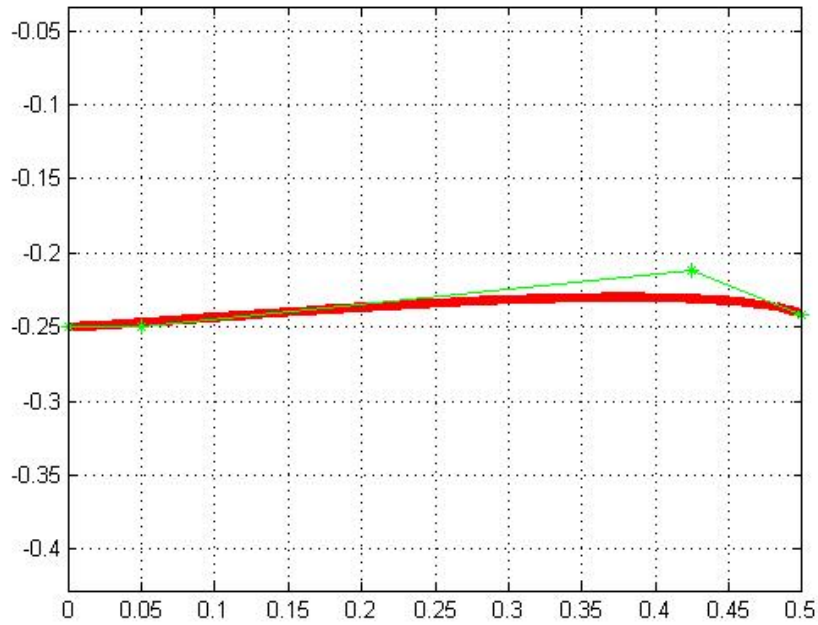


Figure 3.46 Detailed geometry of Bezier curve #4 and its control points

3.3.3 Optimization Results

After implementing MOGA for 20 generations with 8 individuals in each generation, an optimal shape for asymmetric Busemann-type airfoil under lifting condition with maximum c_l and minimum c_d is obtained. Figure 3.47 shows the geometry of the original Busemann biplane airfoil (red) and the optimized Busemann biplane airfoil (blue) under lifting condition. Figures 3.48 - 3.51 show the four Bezier curves used to generate the optimized Busemann biplane airfoil shape along with their control points.

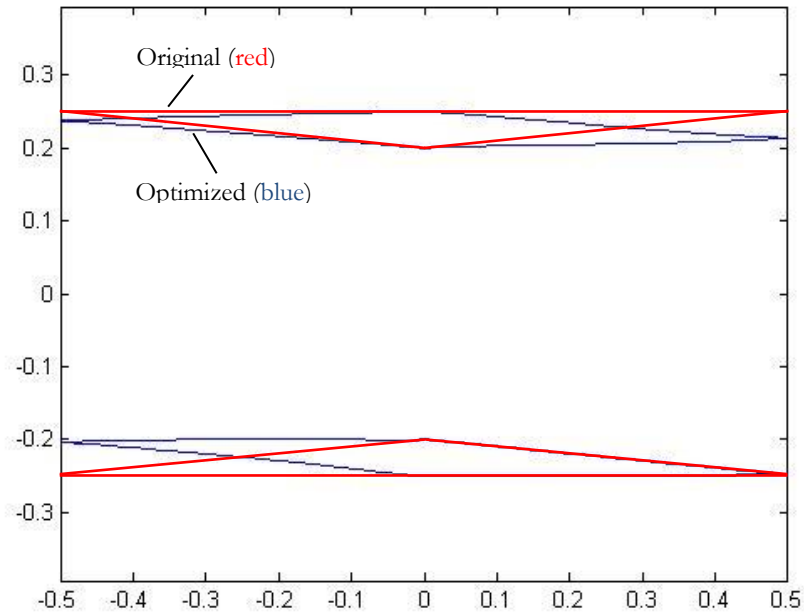


Figure 3.47 Geometry of both the original and optimized Busemann biplane airfoil under lifting condition

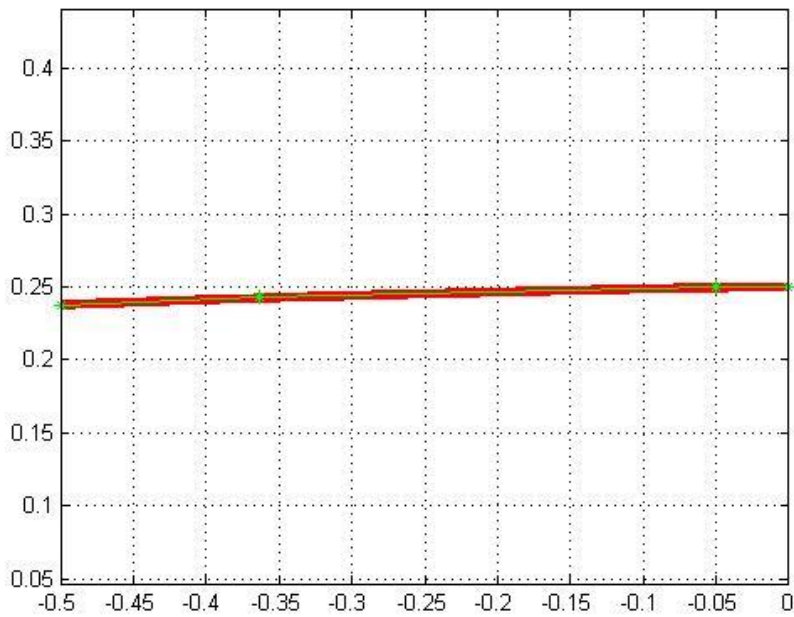


Figure 3.48 Bezier curve #1 and its control points for the optimized Busemann biplane airfoil under lifting condition

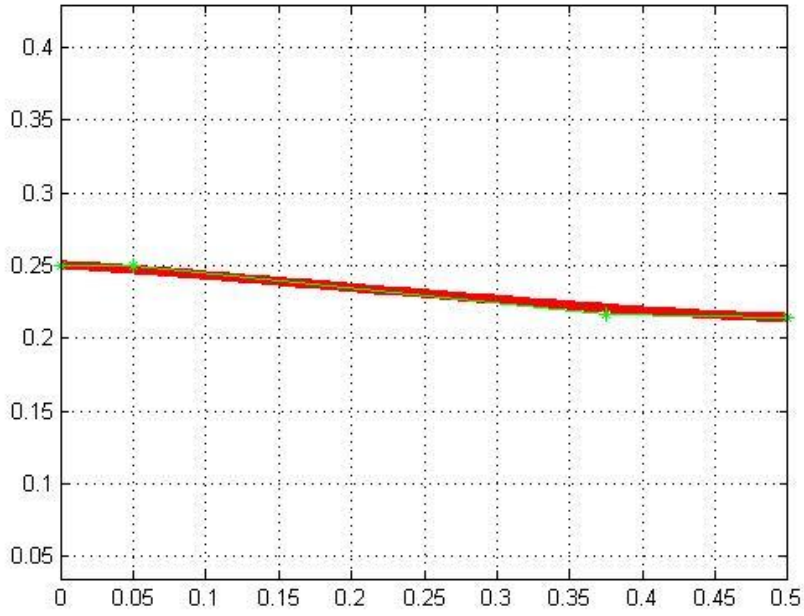


Figure 3.49 Bezier curve #2 and its control points for the optimized Busemann biplane airfoil under lifting condition

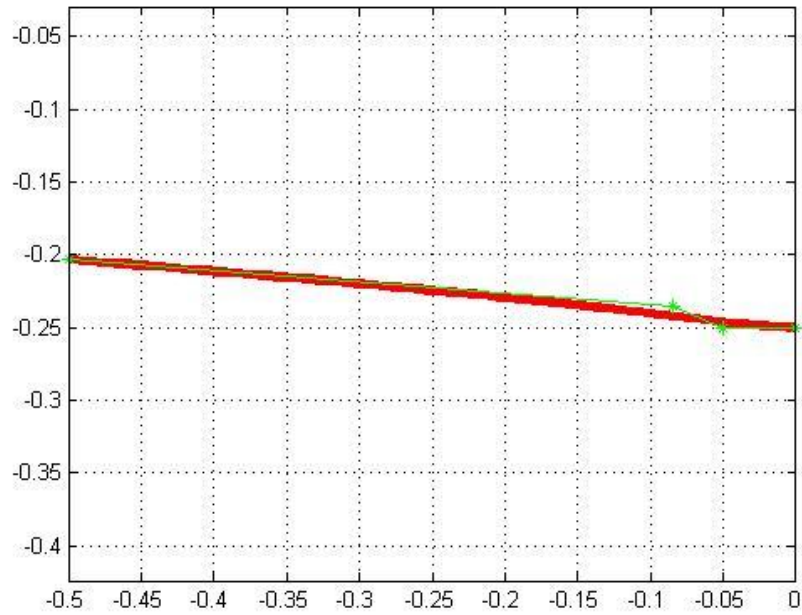


Figure 3.50 Bezier curve #3 and its control points for the optimized Busemann biplane airfoil under lifting condition

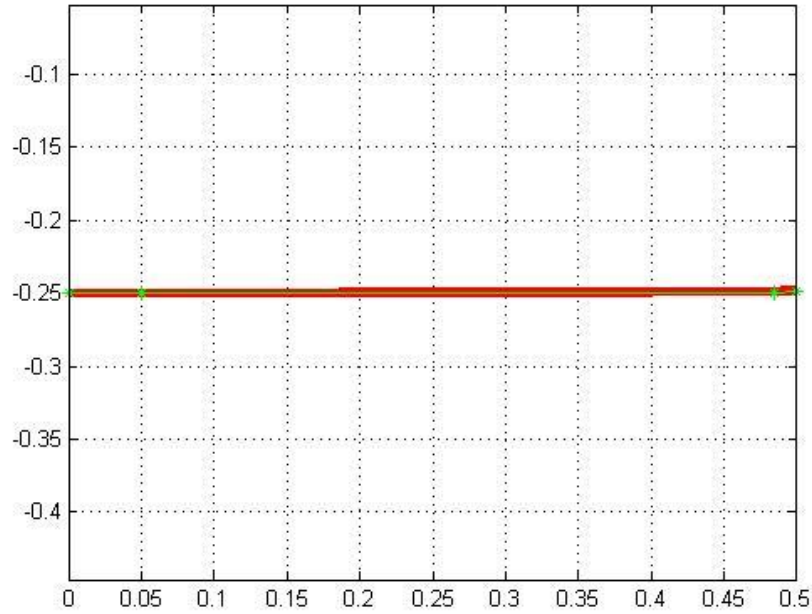


Figure 3.51 Bezier curve #4 and its control points for the optimized Busemann biplane airfoil under lifting condition

The drag coefficients for the seven design points are compared in Table 3.10 and Table 3.11 for both the original and optimized Busemann biplane airfoil under lifting condition. As shown in Table 3.10 and Table 3.11, the baseline Busemann biplane airfoil is choked at all Mach numbers within the optimization range; while the optimized Busemann biplane airfoil unchokes at $M = 1.6$ during acceleration and chokes at $M = 1.3$ during deceleration. Even under choked conditions during both acceleration and deceleration, the optimized Busemann biplane airfoil has significantly lower drag compared to the baseline Busemann biplane airfoil. Similar to the nonlifting condition, the only point where the optimized Busemann biplane airfoil has a higher drag compare to the original airfoil is at $M = 1.7$ during deceleration.

Table 3.10 c_d comparison for the original and optimized Busemann biplane airfoil at lifting condition during acceleration (1 count = 0.0001)

Mach Number	1.1	1.2	1.3	1.4	1.5	1.6	1.7
Baseline	1097	1040	999	967	943	926	923
Optimized	683	633	529	461	418	148	136

Table 3.11 c_d comparison for the original and optimized Busemann biplane airfoil at lifting condition during deceleration (1 count = 0.0001)

Mach Number	1.7	1.6	1.5	1.4	1.3	1.2	1.1
Baseline	32	926	940	967	999	1040	1098
Optimized	136	148	173	224	529	633	682

Figures 3.52 – 3.65 show the change in the pressure coefficient c_p around the optimized Busemann biplane airfoil under lifting condition as the Mach number increases and decreases within the design-point range. As can be seen from these figures, the bow shock wave in front of the airfoil disappears at $M = 1.6$ during acceleration and is not generated until the flow speed drops down to $M = 1.3$ during deceleration.

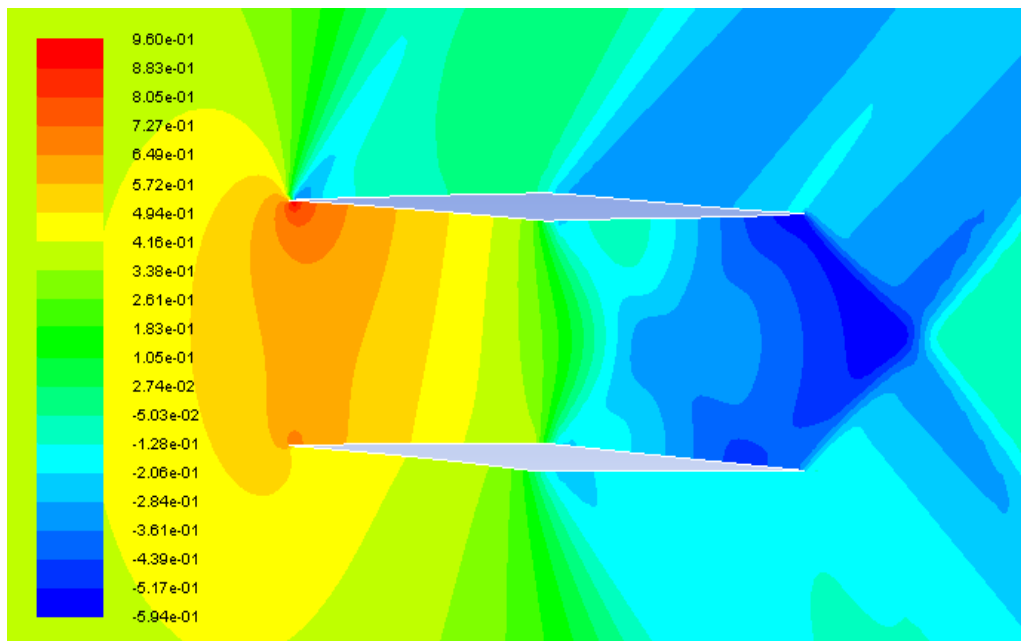


Figure 3.52 c_p contours for the optimized Busemann biplane airfoil under lifting condition at $M = 1.1$ during acceleration

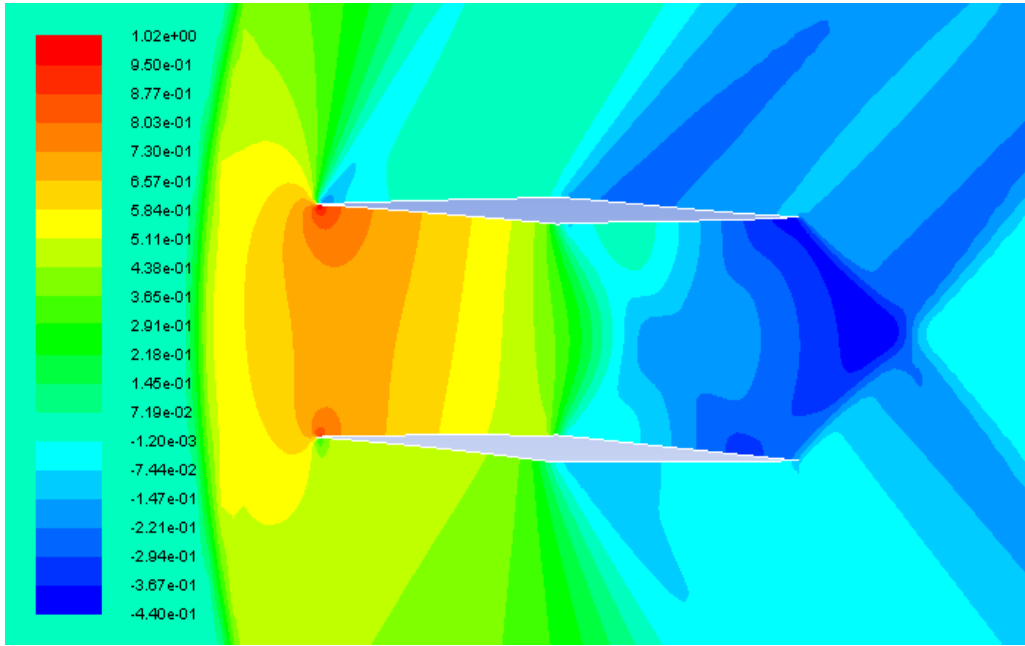


Figure 3.53 c_p contours for the optimized Busemann biplane airfoil under lifting condition at $M = 1.2$ during acceleration

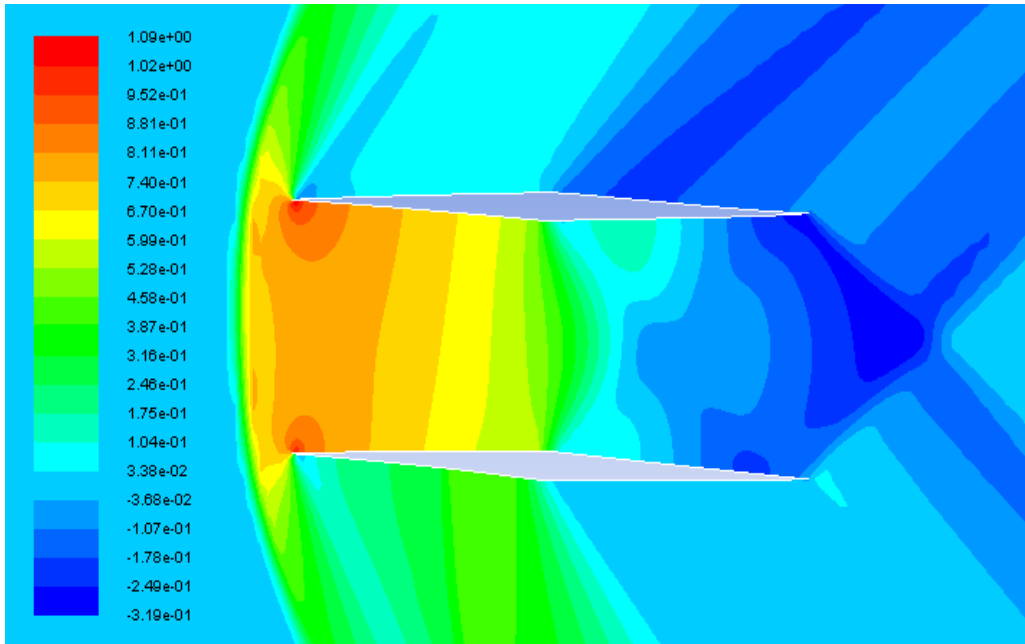


Figure 3.54 c_p contours for the optimized Busemann biplane airfoil under lifting condition at $M = 1.3$ during acceleration

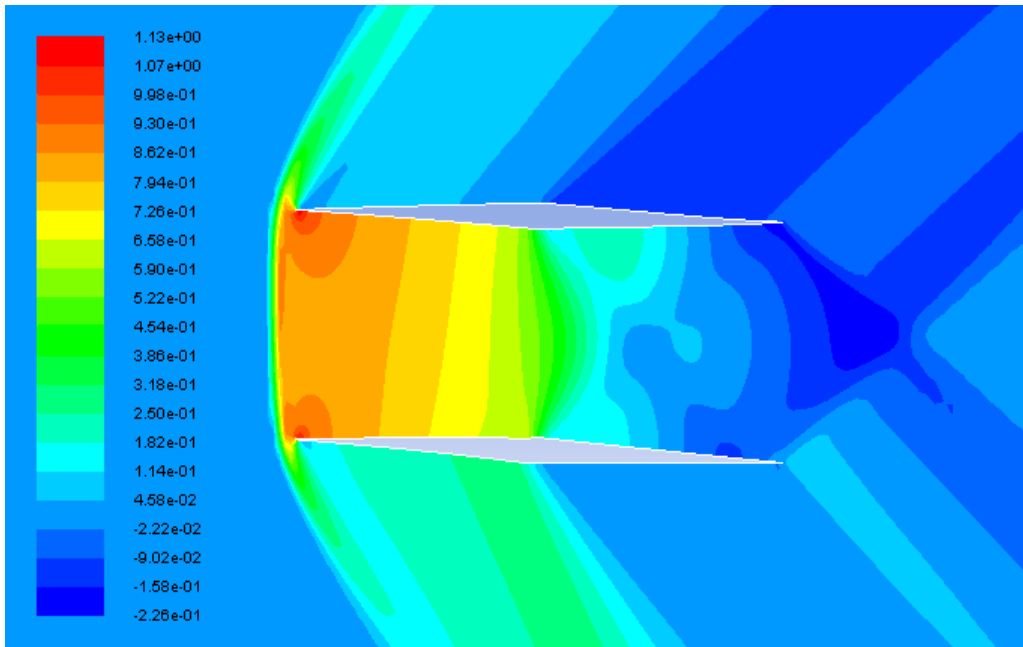


Figure 3.55 c_p contours for the optimized Busemann biplane airfoil under lifting condition at $M = 1.4$ during acceleration

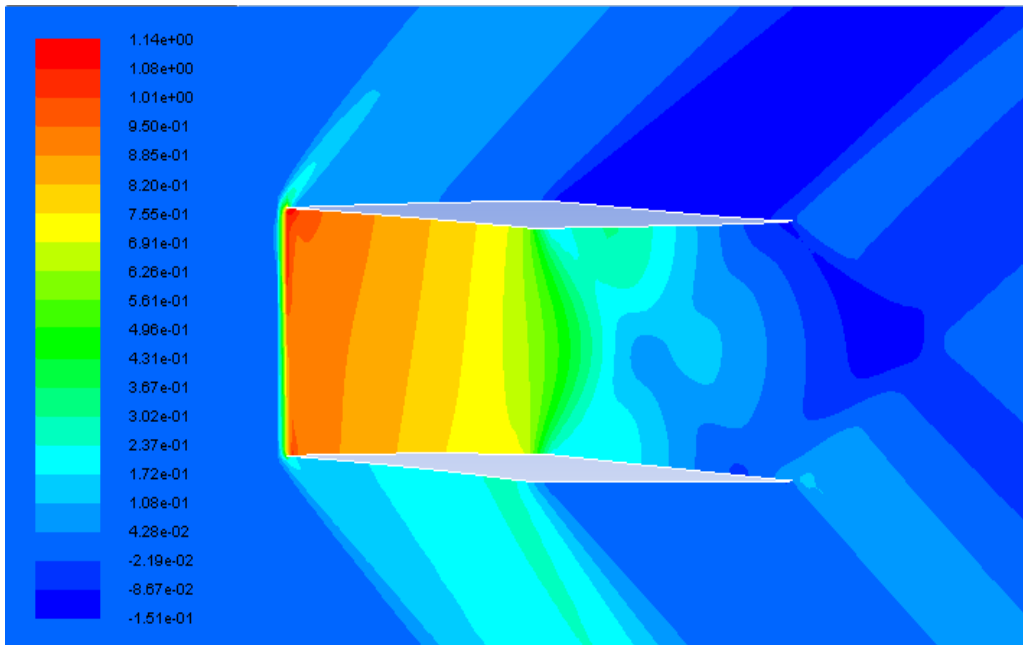


Figure 3.56 c_p contours for the optimized Busemann biplane airfoil under lifting condition at $M = 1.5$ during acceleration

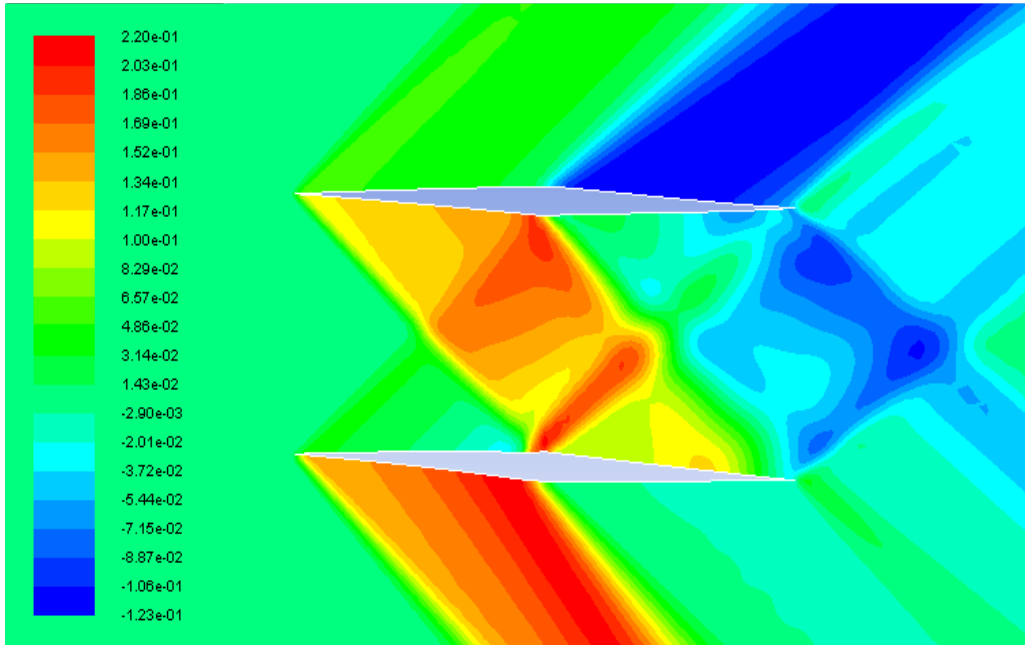


Figure 3.57 c_p contours for the optimized Busemann biplane airfoil under lifting condition at $M = 1.6$ during acceleration

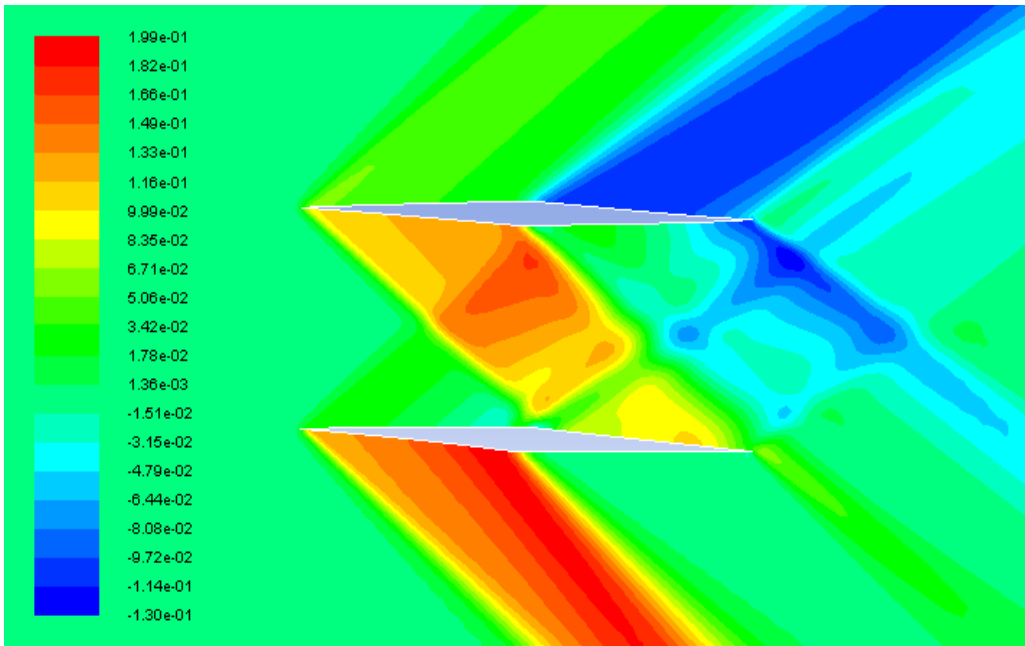


Figure 3.58 c_p contours for the optimized Busemann biplane airfoil under lifting condition at $M = 1.7$ during acceleration

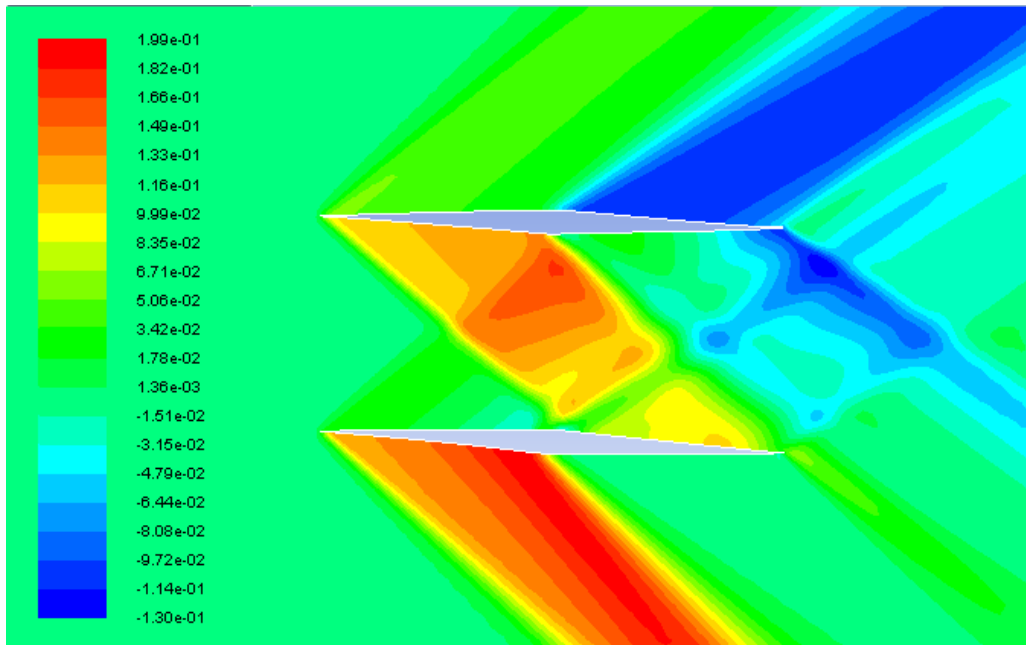


Figure 3.59 c_p contours for the optimized Busemann biplane airfoil under lifting condition at $M = 1.7$ during deceleration

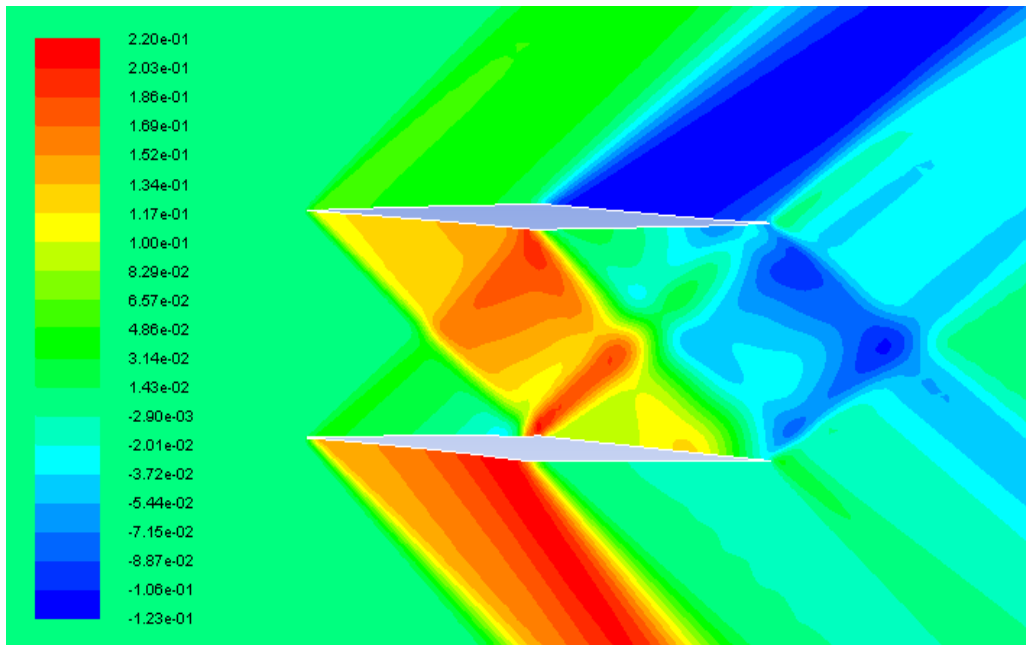


Figure 3.60 c_p contours for the optimized Busemann biplane airfoil under lifting condition at $M = 1.6$ during deceleration

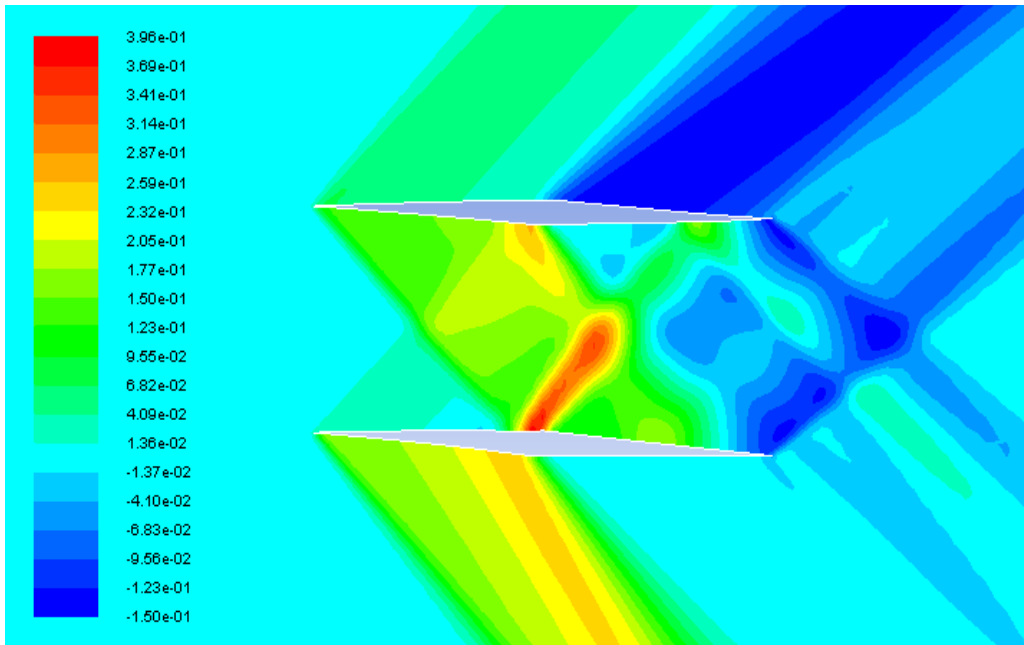


Figure 3.61 c_p contours for the optimized Busemann biplane airfoil under lifting condition at $M = 1.5$ during deceleration

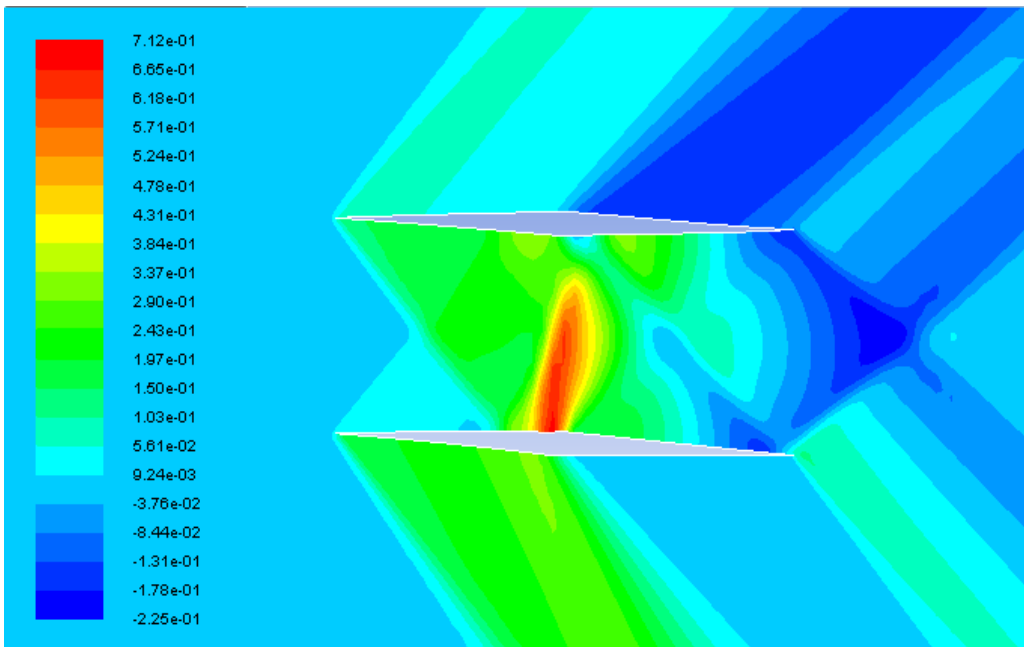


Figure 3.62 c_p contours for the optimized Busemann biplane airfoil under lifting condition at $M = 1.4$ during deceleration

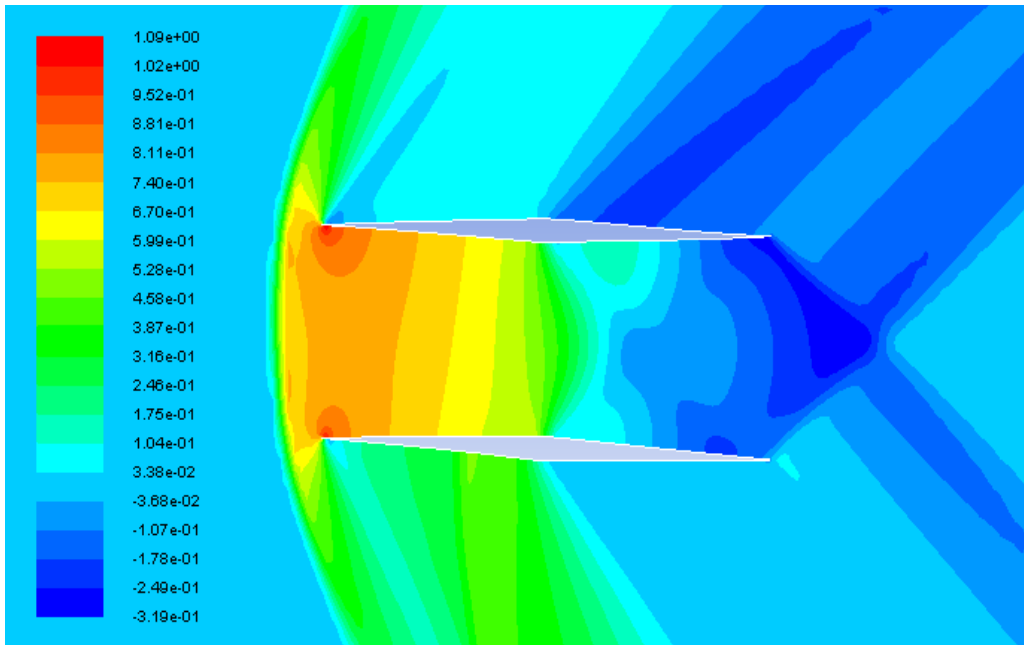


Figure 3.63 c_p contours for the optimized Busemann biplane airfoil under lifting condition at $M = 1.3$ during deceleration

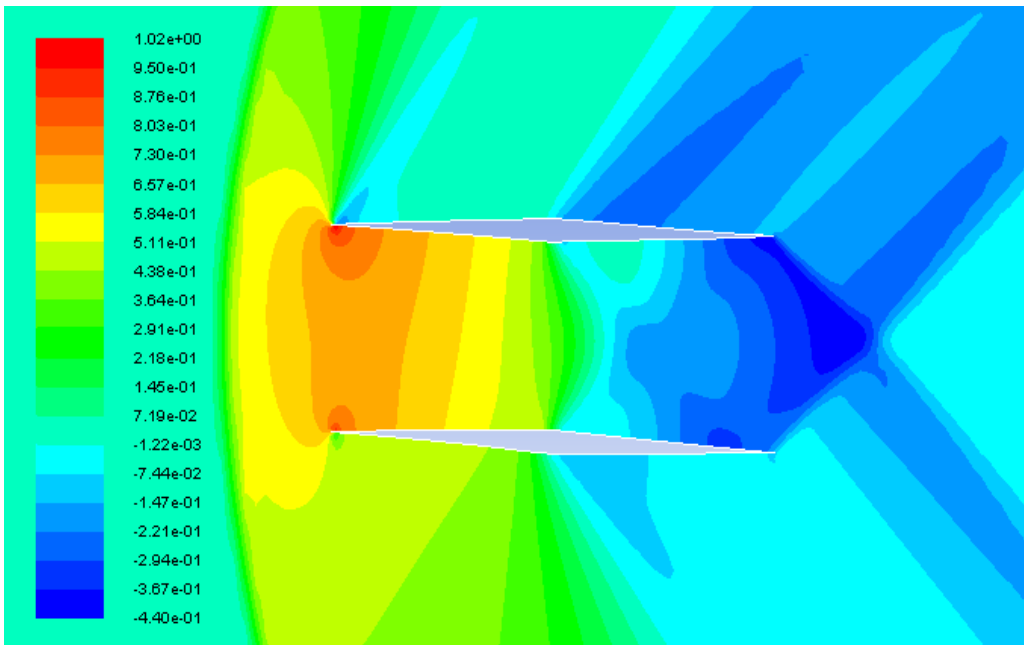


Figure 3.64 c_p contours for the optimized Busemann biplane airfoil under lifting condition at $M = 1.2$ during deceleration

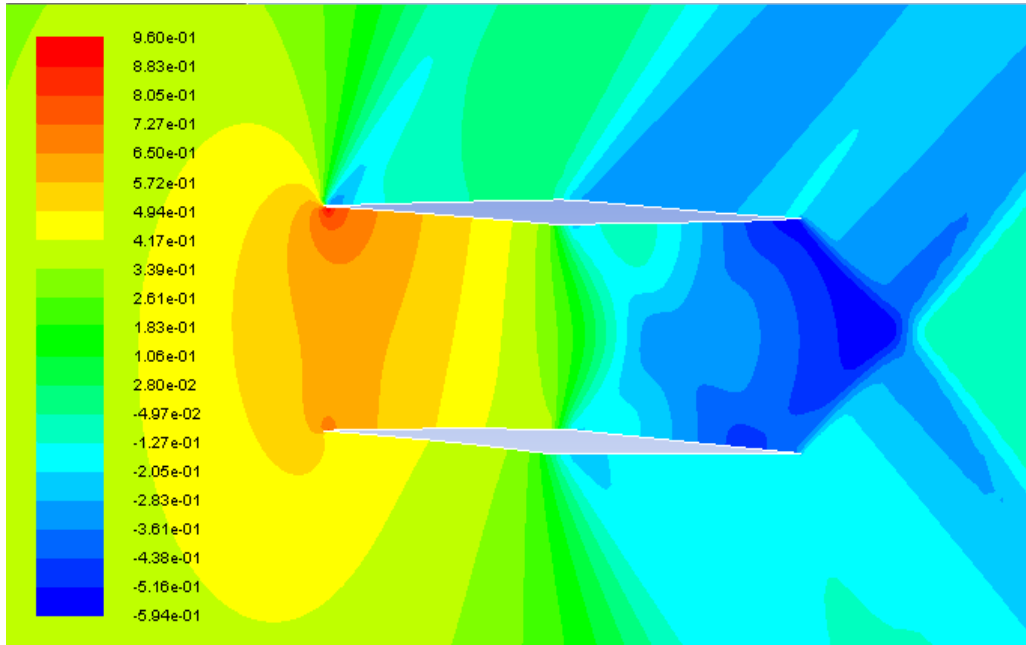


Figure 3.65 c_p contours for the optimized Busemann biplane airfoil under lifting condition at $M = 1.1$ during deceleration

Figure 3.66 shows the comparison of the drag coefficients for the standard diamond-shaped airfoil, the baseline Busemann biplane airfoil and the optimized Busemann biplane airfoil under lifting condition. As shown in the figure, the separation between the acceleration and deceleration c_d lines still exists for the optimized Busemann biplane airfoil, which means that the flow-hysteresis and the choked-flow effects are not totally eliminated. However, as clearly shown in Figure 3.66, the flow-hysteresis area has been significantly reduced and the drag increase during deceleration due to the choked-flow phenomenon is much smaller than that for the original Busemann biplane airfoil. The drag of the optimized Busemann biplane airfoil in the subsonic region is smaller than that of the original Busemann biplane airfoil, although it is slightly higher than that of the standard diamond-shaped airfoil for $0.3 < M < 0.85$. For both subsonic and supersonic conditions, the optimized Busemann biplane airfoil has significantly reduced wave drag compared to the original Busemann biplane airfoil. Similar to the nonlifting condition, the drag coefficient of the optimized Busemann biplane airfoil at $M = 1.7$ ($c_d = 0.01362$) is higher than that of the original Busemann biplane airfoil ($c_d = 0.002182$).

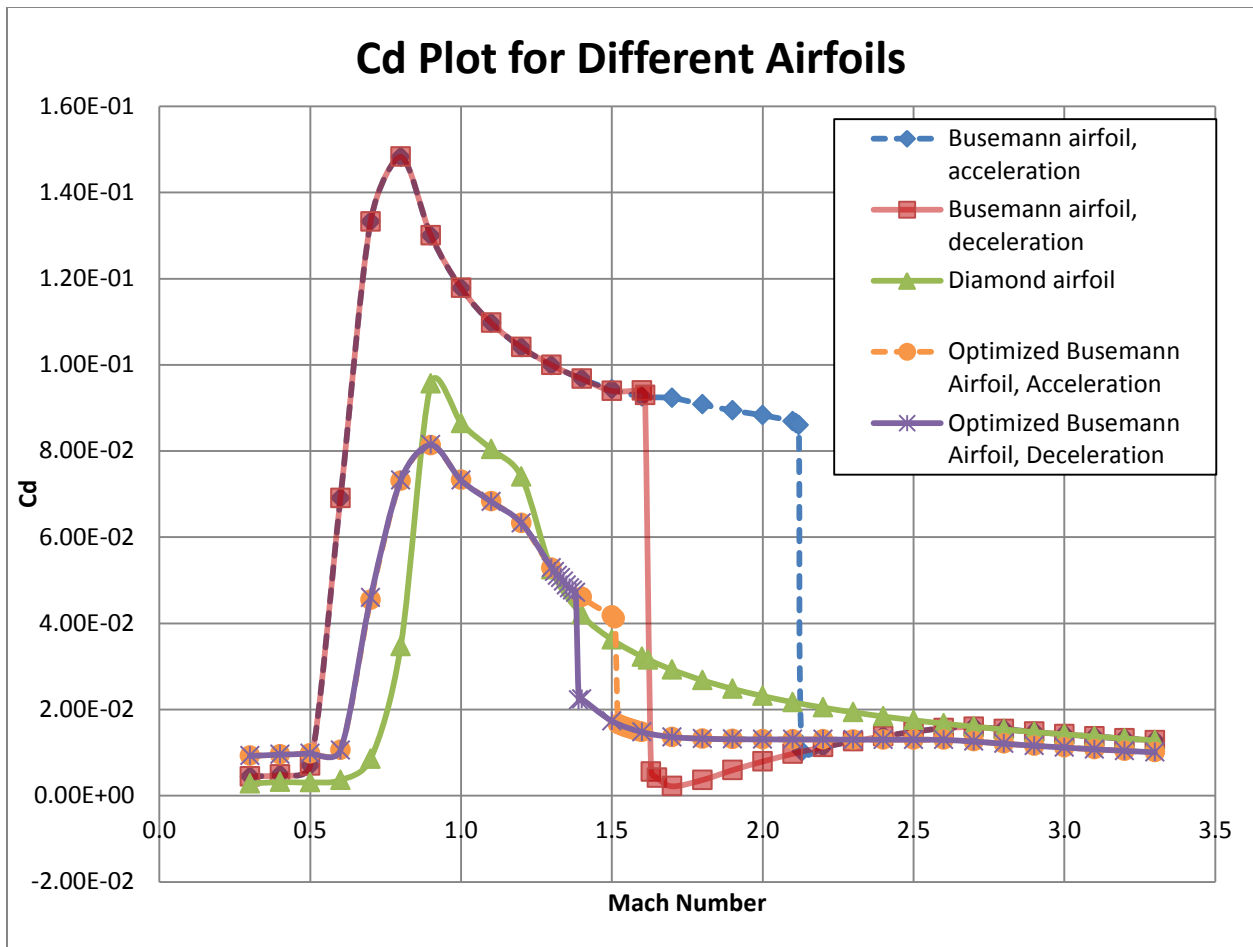


Figure 3.66 c_d plot for different airfoils under lifting condition

Next we examine the details of the flow field in acceleration and deceleration conditions for the optimized Busemann biplane airfoil under lifting condition and compare them with the flow field of the original Busemann biplane airfoil. Figure 3.67 and Figure 3.68 show the pressure coefficient contours of the optimized Busemann biplane airfoil using GA under acceleration and deceleration respectively. During acceleration, the flow-hysteresis effect still exists and a bow shock wave is formed in front of the airfoil. The swallowing of the bow shock wave happens when the Mach number increases from 1.51 to 1.52 as shown in Figure 3.37(f) and from 2.12 to 2.13 for the original Busemann biplane airfoil as shown in Figure 2.16(f). The corresponding drag coefficient decreases from 0.04116 to 0.01665 for the optimized Busemann biplane airfoil.

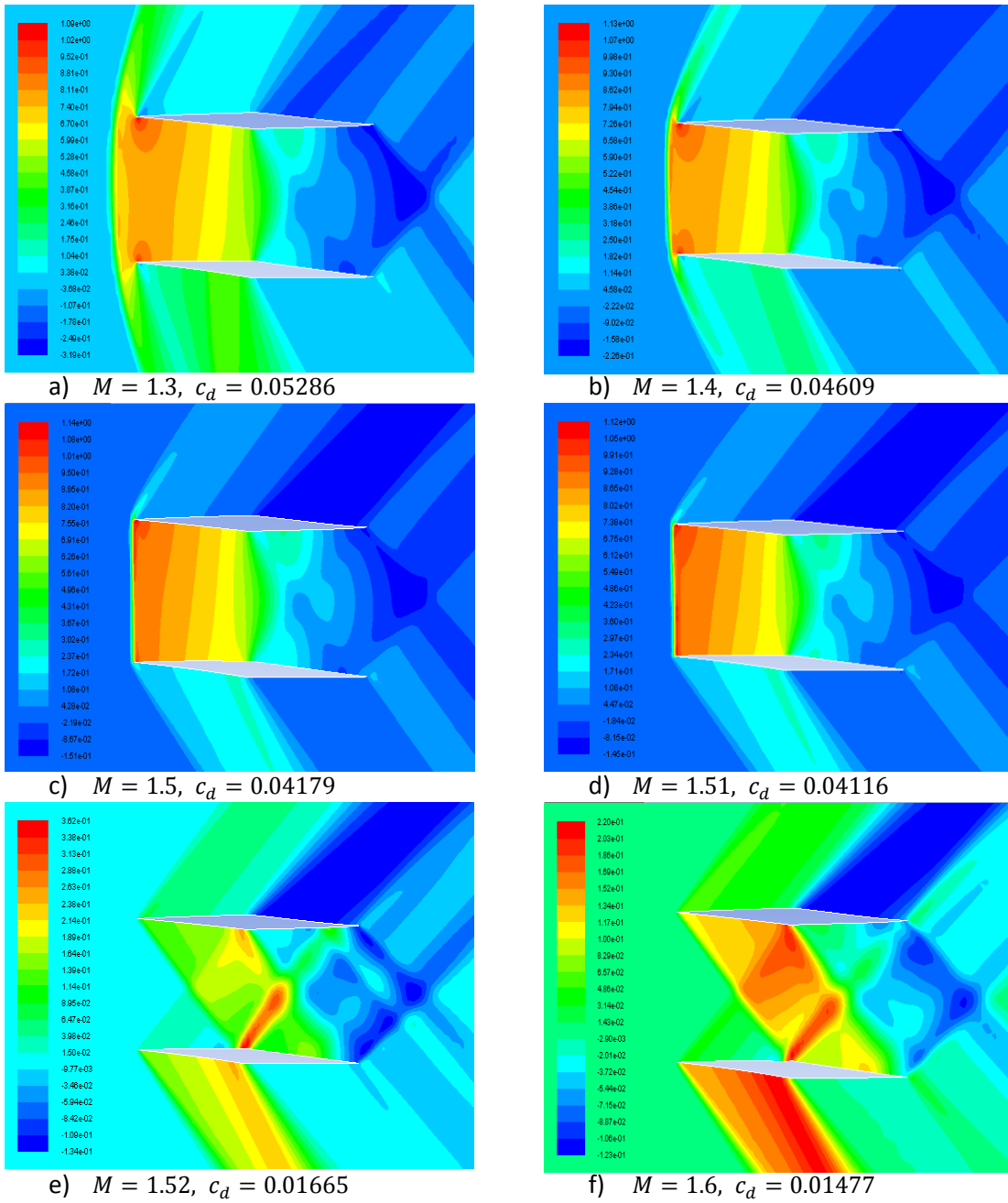


Figure 3.67 c_p -contours of the GA-optimized Busemann biplane airfoil with lift during acceleration

During deceleration, the choked-flow effect still exists; however, it is shifted to a lower Mach number of 1.38 compared to 1.6 for the original Busemann biplane airfoil.

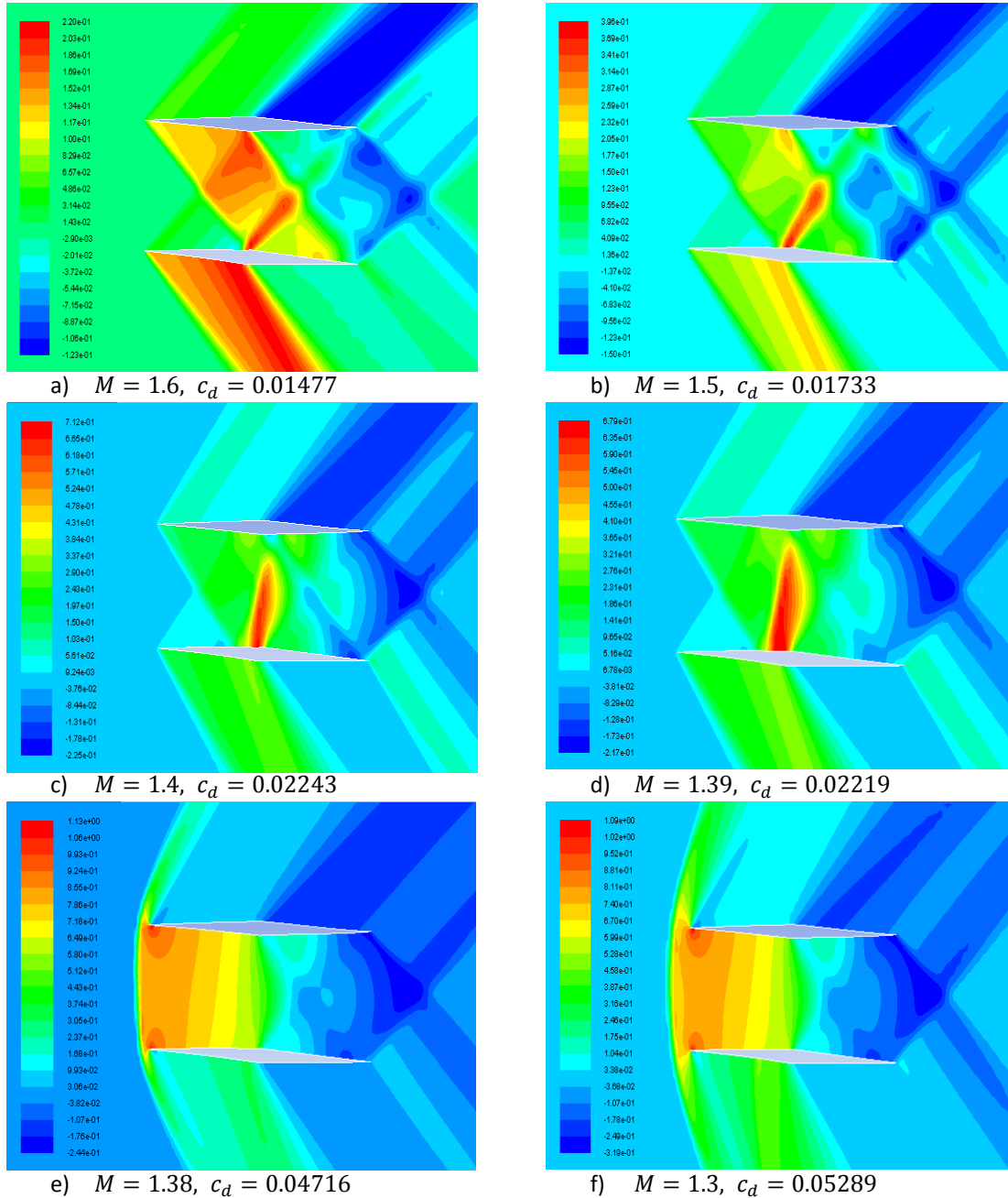


Figure 3.68 c_p -contours of the GA-optimized Busemann biplane airfoil with lift during deceleration

In conclusion, the drag coefficient of the optimized Busemann biplane airfoil under lifting condition is significantly reduced compared to the original Buseman biplane airfoil as shown in Table 3.12.

Table 3.12 c_l for the optimized Busemann biplane airfoil under lifting condition at different design points

Mach Number	1.1	1.2	1.3	1.4	1.5	1.6	1.7
c_l (acceleration)	0.3319	0.3553	0.3061	0.2842	0.2858	0.2525	0.2713
c_l (deceleration)	0.3306	0.3551	0.3064	0.3032	0.2678	0.2525	0.2712

Chapter 4

Conclusions

In this thesis, numerical simulations of the flow past the standard diamond-shaped airfoil and the baseline Busemann-type biplane airfoil have been conducted. An impulsive uniform flow, a flow during acceleration and a flow during deceleration are simulated. The original Busemann biplane airfoil has a poor performance at off-design conditions due to the flow-hysteresis phenomenon during acceleration and the choked-flow phenomenon during deceleration. For shape optimization, a Single-Objective Genetic Algorithm (SOGA) and a Multi-Objective Genetic Algorithm (MOGA) are employed to optimize the shape of the Busemann-type biplane airfoil under nonlifting and lifting condition respectively to improve its performance at off-design conditions. The commercially available CFD solver FLUENT is employed to calculate the flow field on an unstructured mesh generated using the ICEM software. A second-order accurate steady density-based solver in FLUENT is employed to compute the supersonic flow field. The optimization results for the nonlifting case show significant improvement in reducing the drag coefficient at off-design conditions for the optimized Busemann-type biplane airfoil shape compared to the original shape. The flow-hysteresis phenomenon during acceleration and the choked-flow phenomenon during deceleration are both alleviated significantly for the optimized shape. For the lifting case, the optimized Busemann biplane airfoil is able to significantly reduce the drag coefficient at off-design conditions while generating lift at the same time.

Appendix A

Mother Function for Optimization Code

```
% This is a simple multi-objective optimization code that can be used for
% CFD projects here in the lab. It can be adjusted for single or
% dual-objective optimization, and is fully automated. Make sure you know
% the basics of a GA and have some familiarity with matlab before using
% this program. I'm going to try and comment it the best I can. THIS IS A
% GENERAL CODE AND HAS MULTIPLE COMPONENTS THAT MUST BE SPECIALIZED FOR YOUR
% PARTICULAR CASE. I've included "FilesToBeIndividualized.txt" that lists
% all those files. All functions and scripts are commented to help you
% follow what they are doing, but you should only have to change the
% particular files listed.

% This specific matlab file, "MOTHER", is the parent function from which
% everything else is called.

% ----- Begin Code ----- %

%-----
% Step 1: Defining parameters for use throughout simulation
%-----
% The following functions start your program by defining key parameters for
% use throughout the code. Of these, GAparams and GeometryConstraints need
% to be completely individualized for your code. Instructions on how to
% individualize them can be found in their respective .m files.

% 1a) Run "StartOrRestart" to define initialization info
StartOrRestart;

% 1b) Run "GAparams" to define parameters for the GA
GAparams

% Run "GeometryConstraints" to define the specific geometry constraints and
% information this individual case
[Mcurves,Ncurves,nBC,fCPinfo,vCPinfo,limCPinfo] = GeometryConstraints;

% Run "GlobalData" to define our shape and design variable cell arrays, but
% only if it is the initial run.
if strcmp(initRun,'Y') == 1
    [BEZIER_POINTS, DAUGHTER_CURVES, DES_VARS, OLD_DES_VARS] =
GlobalData(genSize,nBC,Ncurves,Mcurves,nDV);
end

% Run "SimTimes" to define how long to pause for meshing and simulation
```

SimTimes

```
%-----  
% Step 2: Randomized Shapes Loop  
%-----  
% This is the first step for the GA: creating, simulating, and evaluating a  
% generation of random individuals. Once a set of random individuals has  
% been created, it will be saved separately from the others for future use.  
% If you are restarting the GA, as noted in the prompt for "initial run,"  
% this entire step will be skipped and the code will begin at step 4.  
  
if strcmp(initRun,'Y') == 1 || (strcmp(initRun,'Y') == 0 && generation == 1  
&& numTrial <= genSize)  
    RandomShapesLoop  
end  
  
%-----  
% Step 3: Genetic algorithm  
%-----  
% Now that we have our generation of random curves and their data stored in  
% our  
% global matrices, we can run the genetic algorithm (GA) and then repeat for  
% a loop using daughter curves. The genetic algorithm is simple once you  
% understand it, but calls multiple functions. Read carefully.  
  
if strcmp(initRun,'Y') == 1 || (strcmp(initRun,'Y') == 0 && generation == 1  
&& numTrial == genSize+1)  
    MOGAgeneral  
    ResetVars  
end  
  
%-----  
% Step 4: Loop until converged  
%-----  
% After the above 3 steps we now have our cell array "BEZIER_CURVES"  
% containing all of the CP information for the newly generated curves. We  
% also have saved the best curve's information. Now we cycle through a very  
% similar loop to "RandomShapeLoop", except our "RandomIndividuals"  
% function is replaced with something that is no longer random. Everything  
% else is, for the most part, the same.  
  
if maxGen ~= 1;  
    while ConvergenceCheck(bestCurveVals,nDV,maxGen) == 0 % while the  
solution is not converged  
        OptimizationLoop        % run the in-generation loop  
        MOGAgeneral  
        ResetVars  
    end  
end  
totalRuns = generation*genSize %#ok<NOPTS>  
  
%-----  
% end of script
```

Appendix B

GAparams.m

```
% GAparams is a script to take in your parameters for the GA. It needs to  
% be individualized for your case.
```

```
% -----  
% Step 1: Individuals per generation and maximum generations  
% -----
```

```
% 1a) Define generation size (number of individuals per generation)  
genSize = 8; %ind % must be greater than 1 and an even integer.
```

```
% Making sure genSize is a positive and even integer.  
if genSize < 2 || rem(genSize,2) ~= 0 || fix(genSize) ~= genSize  
    error('genSize must be a positive and even integer')  
end
```

```
% 1b) Define maximum generations for the optimization  
maxGen = 50; %ind % must be greater than 1 and an integer
```

```
% Making sure maxGen is a positive integer  
if maxGen < 1 || fix(maxGen) ~= maxGen  
    error('Maximum generations must be a positive integer')  
end
```

```
% -----  
% Step 2: Number of design variables (dVs)  
% -----
```

```
% 2a) Define the number of design variables  
% Number of design variables (nDV) is how many variables you need to keep  
% track of to use in your fitness function.  
nDV = 28; %ind
```

```
% Making sure nDV is a positive integer  
if nDV < 1 || fix(nDV) ~= nDV  
    error('nDV must be a positive integer')  
end
```

```
% 2b) Specify dV objectives  
% List the design objective for each dv. If you are trying to minimize  
% them, input "'min'", if you are trying to maximize them, input "'max'"  
% (these inputs are without the " " but do include the ' ' )  
dvObj = cell(1,nDV); % allocates space
```

```
for i = 1:14
```

```

        dvObj{i} = 'min'; %ind
end
for i = 15:28
    dvObj{i} = 'max';
end

% DVerorCheck ensures you input the dV objectives correctly
DVerorCheck(dvObj,nDV)

% -----
% Step 3: Selection Type
% -----
% I have coded in roulette wheel selection and tournament selection.
% You define which type and its corresponding probability values here.
% Use 'rw' for roulette wheel and 't' for tournament (you need to
% apostrophes for these definitions)

% selectionType determines which type of selection will happen
selectionType = 'rw'; %ind
if strcmp(selectionType,'rw') == 1           % if "rw" do roulette wheel
selection
    fprintf('Roulette Wheel Selection\n')
    k = 'not used';
elseif strcmp(selectionType,'t') == 1
    fprintf('Tournament Selection\n')       % if "t" do tournament selection

    % Define tournament size (wealer individuals have a higher probability
    % of being selected as the size becomes larger. When k = 1 then this
    % behaves as random selection, which is bad so don't do it.

    k = 2;      %ind % k can range from 1 to genSize

    % k must be a positive integer less than genSize
    if k < 1 || k > genSize || fix(k) ~= k
        error('"k" value for tournament selection must be a positive integer
less than genSize')
    end
else
    % Make sure selection type is one of the options
    error('"selectionType" is not a viable option. (must be ''t'' or ''rw'')')
end

% -----
% Step 4: Crossover and mutation parameters
% -----
% 4a) Crossover parameter
% This is the probability that crossover occurs for any given variable
control
% point. It must be between 0 and 1.
crossRate = .7; %ind

% makes sure the rate is an acceptable number

```

```

if crossRate >= 1 || crossRate <= 0
    error('Crossover rate is outside allowable bounds')
end

% 4b) Mutation rate
% This is the probability that any given variable control point is mutated.
% It helps increase genetic diversity to avoid local optimums. It must be
% between 0 and 1.

mutRate = .1; %ind

if mutRate >= 1 || mutRate <= 0
    error('Mutation rate is outside allowable bounds')
end

% 4c) Error of mutation constant (eomC)
% eomC determines how much mutation affects the curves as generations go
% on. Set to 0 for a ton of mutation variance (not recommended) and to 1
% for very little mutation (not recommended). Basically, as "generation"
% approaches
% "maxGens" the greatest change a mutation can make is 1-eomC*(distance to
% geometric limit).

eomC = .8; %ind

% makes sure eomC is between 0 and 1
if eomC >= 1 || eomC <= 0
    error('Error of mutation constant is outside allowable bounds')
end

%-----
% end of script

```


Appendix C

MOGAgeneral.m

```
% This is a generic framework for an adaptable single objective or multi
% objective genetic algorithm. It is based on the inputs from "Gparams"
% and "GeometryConstraints"

% -----
% Step 1: Use your fitness function to evaluate each shape's fitness
% -----

fitness = FitnessFunction(genSize,DES_VARS);

% -----
% Step 2: Natural selection sorting
% -----

[BEZIER_POINTS,DES_VARS,sortedBPinfo,sortedTrialData,sortedFitness] =
Sorter(generation,genSize,fitness,BEZIER_POINTS,DAUGHTER_CURVES,DES_VARS,OLD_
DES_VARS);

% -----
% Step 3: Save data used for convergence analysis
% -----

% Save best curve as a vector for reference
for i = 1:nBC
    bestCurve{i}(2*generation-1:2*generation,:) = sortedBPinfo{i}(1:2,:);
    %#ok<SAGROW>
    filename = sprintf('%s_%d','bestCurve',i);
    dlmwrite(filename,bestCurve{i},'-append','newline','pc')
end

% Save the best curve dVs as a vector for reference
for i = 1:nDV
    bestCurveVals(generation,i) = sortedTrialData(1,i+1); %#ok<SAGROW>
end

dlmwrite('bestCurveVals.txt',bestCurveVals,'-append','newline','pc');

% -----
% Step 4: Create new generation
% -----

% Backup before
SaveCurves(BEZIER_POINTS,DAUGHTER_CURVES)

% Create new generation
```

```

DAUGHTER_CURVES =
DaughterCurveGenerator(sortedBPinfo,limCPinfo,vCPinfo,sortedFitness,genSize,s
electionType,crossRate,mutRate,DAUGHTER_CURVES,generation,maxGen,eomC,k);

% Define "old" matrices (sorted already)
BEZIER_POINTS = sortedBPinfo;

OLD_DES_VARS = DES_VARS;

% Backup new generation
SaveCurves(BEZIER_POINTS,DAUGHTER_CURVES)
SaveDVs(DES_VARS,OLD_DES_VARS)

% -----
% end of function.

```

Appendix D

FitnessFunction.m

```
function [fitness] = FitnessFunction(genSize,DES_VARS)
% FitnessFunction needs to be individualized for you fitness function. It has
% access to DES_VARS and therefore can find the fitness. Save the fitness
% as a matrix of trial number in column 1 and fitness in column 2, and
% appropriately title it "fitness"

% My fitness function for the example case of minimizing curve length

% Normalized based on a bad (max) value
fitness = zeros(genSize,2);
fitness(:,1) = 1:genSize;
nDV = length(DES_VARS);
localDragHolder = zeros(1,nDV/2);
localLiftHolder = localDragHolder;
averageVals = zeros(genSize,3);
averageVals(:,1) = 1:genSize;

% Felix Code -----
% Drag
for j = 1:genSize
    for i = 1:nDV/2
        localDragHolder(i) = DES_VARS{i}(j,2);
    end

    % Finding any infinity values
    infIndex = find(localDragHolder == inf); %#ok<EFIND>
    if isempty(infIndex) == 1
        averageVals(j,2) = sum(localDragHolder)/(nDV/2);
    else
        averageVals(j,2) = nan;
    end
end

% Lift
for j = 1:genSize
    for i = (nDV/2+1):nDV
        localLiftHolder = abs(DES_VARS{i}(j,2));
    end
```

```

    % Finding any negative infinity values
    infIndex = find(localLiftHolder == -inf); %#ok<EFIND>
    if isempty(infIndex) == 1
        averageVals(j,3) = sum(localLiftHolder)/(nDV/2);
    else
        averageVals(j,3) = nan;
    end
end

% -----
% Drag
minDragVal = min(averageVals(:,2));
maxDragVal = max(averageVals(:,2));
dragValues = averageVals(:,2);
infIndex = find(isnan(dragValues) == 1);
dragValues(infIndex) = maxDragVal; %#ok<FNDSB>
averageVals(:,2) = dragValues;
normalizedDrag = minDragVal./dragValues*100;

% Lift
maxLiftVal = max(averageVals(:,3));
minLiftVal = min(averageVals(:,3));
liftValues = averageVals(:,3);
infIndex = find(isnan(liftValues) == 1);
liftValues(infIndex) = minLiftVal; %#ok<FNDSB>
averageVals(:,3) = liftValues;
normalizedLift = liftValues./maxLiftVal*100;

% Pareto Optimal front weight (increases/decreases weight of design vars)
weight = .5;
if weight < 0 || weight > 1
    error('Weight is outside acceptable bounds');
end

fitness(:,2) = weight*normalizedLift + (1-weight)*normalizedDrag;

end

% -----
% end of function

```

Appendix E

GenerateIndividuals.m

```
function [BEZIER_POINTS] =  
GenerateIndividuals(numTrial,Mcurves,BEZIER_POINTS)  
% GenerateIndividuals behaves largely the same as RandomIndividuals did,  
% however, instead of creating randomized shapes it takes in curve data  
% (BEZIER_POINTS) and generates the required curve according to numTrial.  
% This curve's point data is saved for being passed to ICEM.  
  
% If you want to graph each individual BC, set plotter = 1;  
% If you want to graph all the BCs together, set plotterAll = 1;  
% Be sure to set both to 0 before running the full GA. You don't want to  
% come back to 400 graphs and your computer crashing.  
  
plotter = 0; %ind  
plotterAll = 0; %ind  
  
%-----  
% Step 1: Start of the loop and limit the geometry randomized inputs  
%-----  
nBC = length(Mcurves); % number of Bezier Curves (BCs) to iterate over  
BPres = cell(1,nBC); % save points for later  
  
for cNum = 1:nBC % for each BC  
  
    j = 0; % counter variable  
    n = Mcurves(cNum)-1; % used for Bezier Point (BP) function  
  
    % Placeholder the BCs as zeros to be filled in later  
    BP = BEZIER_POINTS{cNum}(2*numTrial-1:2*numTrial,:); % gives our current  
    curve  
  
%-----  
% Step 2: Bezier Function Generator  
%-----  
% This is the Bezier Function generator. Go to wikipedia Bezier Curves and  
% you'll find this function under the equations section. THIS FUNCTION DOES  
% NOT CHANGE EVER AND IS SOLELY DEPENDENT ON BPx, BPy, AND THE COUNTING  
% VARIABLES. IT WILL WORK FOR ANY INPUT CURVES. DON'T MESS WITH IT UNLESS  
% YOU KNOW WHAT YOU'RE DOING.  
  
%-----KEEP THIS AS IS-----  
nPoints = 100;  
ux = zeros(1,nPoints);  
uy = zeros(1,nPoints); % these are going to be the point files for ICEM  
uylow = uy;
```

```

for t=0:1/(nPoints-1):1
    j=j+1;
    for i=0:n
        if i==0
            a=(1-t)^n;
        elseif i==n
            a=t^n;
        else
            a=(factorial(n)/(factorial(i)*factorial(n-i)))*t^(i)*(1-
t)^(n-i);
        end
        ux(j)=ux(j)+BP(1,(i+1))*a;
        uy(j)=uy(j)+BP(2,(i+1))*a;

        if cNum == 1
            uylow(j)=uy(j)-0.05-0.1*ux(j);
        elseif cNum == 2
            uylow(j)=uy(j)-.05+.1*ux(j);
        elseif cNum == 3
            uylow(j) = uy(j) +.05+.1*ux(j);
        elseif cNum == 4
            uylow(j) = uy(j) +.05-.1*ux(j);
        end

    end
end
%-----KEEP THIS AS IS-----
%-----
% Step 3: Plotting
%-----
% Plotting the graphs is useful for checking the results in the beginning,
% but be sure to make plotter = 0 at the start of the function before
% running the optimization program.

if plotter == 1
    figure
    plot(ux,uy,'-r','LineWidth',4) % this plots the resulting curve
    hold on
    plot(BP(1,:),BP(2:,:), '*-g') % this plots the Bezier Points
    hold on
    axis equal
    grid on
end

%-----
% Step 4: Saving Shape Data to External Files and Global Matrices
%-----
% Saving the points to a file, Points_#.txt, lets us easily load them into
ICEM.
% See "HelpIcemAutoScript.txt" for some tips on how to most easily do so.

```

```

% Saving the CPs to a global matrix lets us call them later. We will also be
% saving the CPs to an external file for data back up: if for any reason the
% simulation shuts down, the most recent curve information will be
% available in a text file.

% Felix stuff -----

    uxlow = ux;

    points = [ux;uy;zeros(1,length(ux))];
    filename = sprintf('%s_%d', 'PointsBC', cNum);
    fid = fopen(filename, 'w');
    fprintf(fid, '%5.0f %1.0f\n', length(ux), 1);
    fprintf(fid, '\t\t%f\t%f\t%f\n', points);
    fclose(fid);

    points = [uxlow;uylow;zeros(1,length(uxlow))];
    filename = sprintf('%s_%d', 'PointsLow', cNum);
    fid = fopen(filename, 'w');
    fprintf(fid, '%5.0f %1.0f\n', length(ux), 1);
    fprintf(fid, '\t\t%f\t%f\t%f\n', points);
    fclose(fid);
% -----

% This saves the BC CPs to the global BEZIER_POINTS cell array
    MatBeingUsed = BEZIER_POINTS{cNum};
    MatBeingUsed(2*numTrial-1:2*numTrial,:) = BP;
    BEZIER_POINTS{cNum} = MatBeingUsed;

% The formatting means that the CPs for our BC will exist in two rows, one
% for x values and one for y. If it's our first curve they will exist in rows
% 1 (x) and 2 (y). For our 12th curve they will be in rows 23 (x) and 24 (y).
% The size of BEZIER_POINTS was defined in GlobalData using our generation
% size
% and number of control points, so everything should fit perfectly. IF YOU
% HAVE ANY CURVES THAT EXIST IN THREE DIMENSIONS THE ABOVE CODE WILL NOT
% WORK. YOU'LL HAVE TO FIGURE THAT ONE OUT ON YOUR OWN.

    % Save for later
    BPres{cNum} = BP;

end

%-----
% Step 5: Saving results for easy viewing and global plotting.
%-----

% Results.txt is a file that will save your Bezier curve information and
% desVars information for each trial in each generation. We won't access
% any information from it as there are the global matrices and their
% backups, but it does bring all of the important information together into

```

```

% one text file that is easy to read.

% Saving a summary of things for easy viewing
fid = fopen('Results.txt','a');
fprintf(fid, '\nShape %i\n', numTrial);
for i = 1:cNum
    dlmwrite('Results.txt', BPres{i}, '-append')
end
fclose(fid);

% These final lines are used to plot all the BCs together on the same plot.
% To turn off, simply set plotterAll = 0 at the beginning of the code;

PointsBC = cell(1,nBC);
PointsLow = PointsBC;
PointsBCMirror = PointsBC;
PointsLowMirror = PointsBC;
if plotterAll == 1
    figure
    for i = 1:nBC
        % Plotting BCs
        filename = sprintf('%s_%d', 'PointsBC', i);
        PointsBC{i} = dlmread(filename);
        PointsBC{i} = PointsBC{i}(2:end,:); % gets rid of icem point count

        plot(PointsBC{i}(:,1), PointsBC{i}(:,2))
        hold on
        axis equal

        % Plotting low curves
        filename = sprintf('%s_%d', 'PointsLow', i);
        PointsLow{i} = dlmread(filename);
        PointsLow{i} = PointsLow{i}(2:end,:);

        plot(PointsLow{i}(:,1), PointsLow{i}(:,2))
        hold on

    end
end

end

%-----
%end of function

```


Appendix F

DaughterCurveGenerator.m

```
function [DAUGHTER_CURVES] =
DaughterCurveGenerator(sortedBPinfo,limCPinfo,vCPinfo,sortedFitness,genSize,s
electionType,crossRate,mutRate,DAUGHTER_CURVES,generation,maxGen,eomC,k)
% will produce genSize of offspring curves

nBC = length(sortedBPinfo);
delete DaughterCurves.txt
% -----
% Step 1: Selection function
% -----
if strcmp(selectionType,'rw') == 1
    parNums = RouletteSampling(sortedFitness,genSize);
else
    parNums = TournamentSampling(genSize,k);
end

% -----
% Step 2: Crossover of selected curves
% -----
maleCurve = cell(1,nBC);
femaCurve = maleCurve;
for i = 1:genSize % each couple produces 2 kids but we only save 1
    maleNum = parNums(i,1);
    femaNum = parNums(i,2);
    for j = 1:nBC
        maleCurve{j} = sortedBPinfo{j}(2*maleNum-1:2*maleNum,:);
        femaCurve{j} = sortedBPinfo{j}(2*femaNum-1:2*femaNum,:);
    end
    % now we have the male and female curve information and can crossover
    [newCurve] = Crossover(maleCurve,femaCurve,vCPinfo,crossRate);
    for j = 1:nBC
        DAUGHTER_CURVES{j}(2*i-1:2*i,:) = newCurve{j}; % store the info to
DAUGHTER_CURVES
    end
end

% -----
% Step 3: Mutation of randomly selected curves
% -----
%random mutation function
DAUGHTER_CURVES =
Mutator(vCPinfo,limCPinfo,mutRate,genSize,DAUGHTER_CURVES,generation,maxGen,e
omC);

end
```

Appendix G

Crossover.m

```
function [newCurve] = Crossover(male,fema,vCPinfo,crossRate)
% Exchange CP info between the curves

nBC = length(male);
new1 = male;
new2 = fema;

for i = 1:nBC % each Bezier curve
    for j = 1:length(vCPinfo{i}) % only the variable CPs
        doCrossover = (rand(1) <= crossRate); %1 if we will, 0 if we won't
        if doCrossover == 1
            sM = male{i}(vCPinfo{i}(j)); % male value of index j
            sF = fema{i}(vCPinfo{i}(j)); % fema value of index j
            new1{i}(vCPinfo{i}(j)) = sF; % male now has fema value
            new2{i}(vCPinfo{i}(j)) = sM; % fema now has male value
        end
    end
end

% Randomly choose which curve
if rand(1) <= .5
    newCurve = new1;
else
    newCurve = new2;
end

end

% -----
% end of function
```

Appendix H

Mutator.m

```
function [DAUGHTER_CURVES] =  
Mutator(vCPinfo,limCPinfo,mutRate,genSize,DAUGHTER_CURVES,generation,maxGen,eomC)  
% Mutator takes in the new daughter curves and iterates over all of their  
% variable CPs. mutRate % of those vCPs will be mutated, or randomly  
% altered.  
  
% -----  
% Step 1: Define the effective mutation efficiency (eom)  
% -----  
% We want eom to start at 1 and decrease to eomC by maxGen  
  
eom = 1 - eomC*generation/maxGen;  
  
% -----  
% Step 2: Mutate the curves  
% -----  
nBC = length(DAUGHTER_CURVES);  
  
for i = 1:genSize % iterate over all individuals  
    for j = 1:nBC % iterate over each bezier curve  
        curve2Mut = DAUGHTER_CURVES{j}(2*i-1:2*i,:); % define curve CPs  
        for k = 1:length(vCPinfo{j}) % iterate over all vCPs  
            doMutation = (rand(1) <= mutRate); % mutation probability is  
mutRate  
            if doMutation == 1  
mutated  
                val2Mut = curve2Mut(vCPinfo{j}(k)); % value of point being  
                if rand(1)<.5  
                    edgeVal = limCPinfo{j}(k,1); % select one edge value  
                else  
                    edgeVal = limCPinfo{j}(k,2); % otherwise select other  
                end  
                curve2Mut(vCPinfo{j}(k)) = val2Mut + eom*rand(1)*(edgeVal-  
val2Mut); %curve moves some random distance towards that edge  
            end  
        end  
        DAUGHTER_CURVES{j}(2*i-1:2*i,:) = curve2Mut; % replacing in DC matrix  
    end  
end  
  
end
```

References

- [1] Busemann, A., *Aerodynamic Lift at Supersonic Speeds*, 12th ed., No. 6, Luftfahrtforschung, Oct 1935, pp. 210-220.
- [2] Moeckel, W. E., “Theoretical Aerodynamic Coefficients of the Two-Dimensional Supersonic Biplane,” NACA Tech. Rept. 1316, 1947.
- [3] Licher, R., “Optimum Two-Dimensional Multiplanes in Supersonic Flow,” Douglas Aircraft Co. Tech Dept. SM-18688, 1955.
- [4] Tan, H. S., “The Aerodynamics of Supersonic Biplanes of Finite Span,” WADC Tech. Rept. 52-276, 1950.
- [5] Ferri, A., *Elements of Aerodynamics of Supersonic Flow*, Macmillan Company, New York, 1949.
- [6] Igra, D., and Arad, E., “A Parametric Study of the Busemann Biplane Phenomena,” *Shock Waves*, Vol. 16, No. 3, Feb. 2007, pp. 269–273.
doi:10.1007/s00193-006-0070-x
- [7] Kusunose, K., Matsushima, K., Goto, Y., Yamashita, H., Yonezawa, M., Maruyama, D., and Nakano, T., “A Fundamental Study for the Development of Boomless Supersonic Transport Aircraft,” AIAA Paper 2006-654, 44rd AIAA Aerospace Sciences Meeting and Exhibit, January 2006.
- [8] Maruyama, D., Matsushima, K., Kusunose, K., and Nakahashi, K., “Aerodynamic Design of Biplane Airfoils for Low Wave Drag Supersonic Flight,” AIAA Paper 2006-3323, 24th Applied Aerodynamics Conference, June 2006.
- [9] Matsushima, K., Maruyama, D., Kusunose, K., and Matsuzawa, T., “Numerical Design and Assessment of a Biplane as Future Supersonic Transport—Revisiting Busemann’s Biplane,” 25th International Congress of Aeronautical Sciences, ICAS Paper 2006-3.7.1, 2006, pp. 1–10.
- [10] Maruyama, D., Matsuzawa, T., Kusunose, K., Matsushima, K., and Nakahashi, K., “Consideration at Off-design Conditions of Supersonic Flows around Biplane Airfoils,” AIAA Paper 2007-687, 45th AIAA Aerospace Sciences Meeting and Exhibit, January 2007.
- [11] Yamashita, H., Yonezawa, M., Obayashi, S., and Kusunose, K., “A Study of Busemann-Type Biplane for Avoiding Choked Flow,” AIAA Paper 2007-688, 45th AIAA Aerospace Sciences Meeting and Exhibit, January 2007.
- [12] Kuratani, N., Ogawa, T., Yamashita, H., Yonezawa, M., and Obayashi, S., “Experimental and Computational Fluid Dynamics around Supersonic Biplane for Sonic-Boom Reduction,”

- AIAA Paper 2007-3674, 13th AIAA/CEAS Aeroacoustics Conference (28th AIAA Aeroacoustics Conference), June 2007.
- [13] Kashitani, M., Yamaguchi, Y., Kai, Y., and Hirata, K., "Preliminary Study on Lift Coefficient of Biplane Airfoil in Smoke Wind Tunnel," AIAA Paper 2008-349, 46th AIAA Aerospace Sciences Meeting and Exhibit, January 2008.
- [14] Maruyama, D., Matsushima, K., Kusunose, K., and Nakahashi, K., "Three-Dimensional Aerodynamic Design of Low-Wave-Drag Supersonic Biplane Using Inverse Problem Method," *Journal of Aircraft*, Vol. 46, No. 6, 2009, pp. 1906-1918.
doi:10.2514/1.40313
- [15] Maruyama, D., Kusunose, K., and Matsushima, K., "Aerodynamic Characteristics of a Two-Dimensional Supersonic Biplane Covering its Take-Off to Cruise Conditions," *Shock Waves*, Vol. 18, No. 6, 2009, pp. 437-450.
- [16] Kusunose, K., Matsushima, K., and Maruyama, D., "Supersonic Biplane," *Progress in Aerospace Sciences*, Vol. 47, No. 1, 2011, pp. 53-87.
doi:10.1016/j.paerosci.2010.09.003
- [17] Hu, R., Jameson, A., and Wang, Q., "Adjoint-Based Aerodynamic Optimization of Supersonic Biplane Airfoils," *Journal of Aircraft*, Vol. 49, No.3, May-June 2012, pp. 802-814.
- [18] 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
- [19] Liepmann, H., and Roshko, A., *Elements of Gasdynamics*, Dover Publications, Inc., Mineola, New York, 1957, pp. 107-118, Ch. Waves in Supersonic Flow.
- [20] He, Yilei, "Shape Optimization of Airfoils Without and With Ground Effect Using a Multi-Objective Genetic Algorithm," M.S. Thesis, School of Engineering and Applied Science, Washington University in St. Louis, 2014.
- [21] Bandyopadhyay, S. and Saha, S., "Some Single- and Multiobjective Optimization Techniques," In *Unsupervised Classification - Similarity Measures, Classical and Metaheuristic Approaches, and Applications*, 17-21, Springer-Verlag Berlin Heidelberg, 2013.
doi: 10.1007/978-3-642-32451-2
- [22] <https://www.math.ubc.ca/~cass/gfx/bezier.html>

Vita

Yi Tian

Degrees M.S. Mechanical Engineering, Washington University in St. Louis, August 2015
B.S. Mechanical Engineering, Saint Louis University, May 2012

Professional Societies American Society of Mechanical Engineers (ASME)

August 2015