



**COMPUTATIONAL OPTIMIZATION UNDER
UNCERTAINTY OF AN ACTIVE FLOW
CONTROL JET**

THESIS

Luke A. Welch, Second Lieutenant, USAF
AFIT-ENY-MS-17-M-299

**DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY**

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

DISTRIBUTION STATEMENT A
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

The views expressed in this document are those of the author and do not reflect the official policy or position of the United States Air Force, the United States Department of Defense or the United States Government. This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

AFIT-ENY-MS-17-M-299

COMPUTATIONAL OPTIMIZATION UNDER UNCERTAINTY OF AN ACTIVE
FLOW CONTROL JET

THESIS

Presented to the Faculty
Department of Aeronautics and Astronautics
Graduate School of Engineering and Management
Air Force Institute of Technology
Air University
Air Education and Training Command
in Partial Fulfillment of the Requirements for the
Degree of Master of Science in Aeronautical Engineering

Luke A. Welch, B.S.M.E.

Second Lieutenant, USAF

March 2, 2017

DISTRIBUTION STATEMENT A
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

AFIT-ENY-MS-17-M-299

COMPUTATIONAL OPTIMIZATION UNDER UNCERTAINTY OF AN ACTIVE
FLOW CONTROL JET

Luke A. Welch, B.S.M.E.
Second Lieutenant, USAF

Committee Membership:

Lt Col Jacob Freeman, PhD
Chair

Dr. Philip Beran, PhD
Member

Dr. Ryan Durscher, PhD
Member

Maj Darrell Crowe, PhD
Member

Abstract

Optimization under uncertainty is performed to determine the optimal parameters of an active flow control jet to impart robust control during transonic cruise. A steady blowing jet is optimized on an NACA 64A-010 airfoil to impart a change in lift greater than or equal to that generated by traditional control surfaces. The design candidates are computationally evaluated using the NASA flow solver, FUN3D, under 20 unique combinations of angle of attack, Reynolds number, and Mach number in order to propagate model input uncertainty. The mean change in lift and the associated standard deviation are included in the optimization framework to help ensure a robust solution. The mass flow rate required to achieve robust control is minimized. Due to time constraints, the optimization failed to produce an optimum solution. However, a number of designs produced an acceptable change in lift to theoretically control an aircraft. One design required a coefficient of mass flow rate of just 1.76×10^{-3} . Translated to a Boeing 747, this is approximately 7.136 kg/s or just 1.14% of the mass flowing through one of its four CF6 engines. The uncertainty associated with the final design is quantified in the form of a probability box. Total predictive uncertainty is estimated as $\pm 40.9\%$, of which $\pm 8.8\%$ is attributed to input uncertainty, $\pm 13.8\%$ to numerical uncertainty, and $\pm 18.3\%$ to model form uncertainty. The input uncertainty results from the fluctuating inputs and is included in the optimization under uncertainty. Further analysis with a refined mesh could greatly decrease numerical uncertainty and a validation experiment could reduce model form uncertainty.

Acknowledgments

I would like to express my deepest gratitude to my thesis advisors, Lt Col Jacob Freeman and Dr. Phil Beran and to my readers, Maj Darrell Crowe and Dr. Ryan Durscher. Their combined insight, experience, and recommendations were invaluable throughout this process. I would also like to thank Dr. Durscher for his contribution of PACMAN, an automatic case management system, which was used heavily throughout this study.

I owe a debt of gratitude to the software developers and support staff associated with FUN3D, Pointwise, and Dakota. Each has contributed greatly to my success.

Finally, my research would not have been possible without the DOD high performance computing resources, Thunder and Spirit, on which the vast majority of my computer simulations were completed.

Luke A. Welch

Contents

	Page
Abstract	iv
Acknowledgments	v
List of Figures	viii
List of Tables	xi
I. Introduction	1
1.1 Motivation and Proposed Solution	1
1.2 Solution Approach	2
1.2.1 Scope	4
1.3 Outcome and Impact	5
1.4 Report Structure	5
II. Background	6
2.1 Chapter Overview	6
2.2 Active Flow Control	6
2.3 AFC Implementation and Design Goals	8
2.4 Computational Fluid Dynamics	10
2.5 Design Optimization	11
2.6 Optimization under Uncertainty (OUU)	14
2.7 Verification, Validation, and Uncertainty Quantification	15
2.7.1 Verification and Numerical Uncertainty	16
2.7.2 Validation and Model Form Uncertainty	18
2.7.3 Total Predictive Uncertainty	20
2.8 Summary	22
III. Research Methodology	23
3.1 Chapter Overview	23
3.2 Sampling Approach	24
3.2.1 Mach Number	25
3.2.2 Reynolds Number	28
3.2.3 Angle of Attack	29
3.2.4 Sampling Summary	30
3.3 Problem Formulation	31
3.3.1 Objective Function	31
3.3.2 Design Variables and Physical Constraints	32
3.3.3 Performance Constraint	34
3.3.4 Optimization Parameters	36
3.3.5 Optimization Problem	37
3.4 Computational Model	37

	Page
3.5 Verification, Validation, and Uncertainty Quantification	44
3.5.1 Numerical Uncertainty	44
3.5.2 Model Form Uncertainty	45
3.5.3 Input Uncertainty	46
3.5.4 Total Predictive Uncertainty	46
IV. Results and Discussion	47
4.1 Optimization	47
4.2 Improved Design	54
4.3 Uncertainty Quantification.....	56
4.3.1 Input Uncertainty	57
4.3.2 Model Form Uncertainty	62
4.3.3 Numerical Uncertainty	64
V. Conclusions and Recommendations	69
Appendix A. Sample FUN3D Input File	72
Appendix B. Dakota Input File: EGO	74
Appendix C. Dakota Input File: Surrogate-based	75
Appendix D. Analysis Driver BASH Script	76
Appendix E. Sample Pointwise Glyph Script	79
Appendix F. Sample PACMAN Input File	81
Appendix G. BASH Script for Generating New Grid	83
Appendix H. BASH Script for Changing FUN3D Input File	84
Bibliography	85

List of Figures

Figure		Page
2.1	Classification of AFC actuators (reproduced from Ref.[5]).	7
2.2	Surrogate-based global optimization.	14
2.3	Example area validation metric between two distinct EDF's. (reproduced from Ref. [18])	20
2.4	Example p-box representing total predictive uncertainty. (reproduced from Ref. [19])	21
3.1	Graphical representation of optimization process.	24
3.2	Drag rise due to wave drag in the transonic flight regime. (reproduced from Ref. [20])	25
3.3	Sampling approach utilized to select cruise Mach numbers.	27
3.4	Graphical representation of each design variable (NACA 64A-010 airfoil).	32
3.5	Baseline finest mesh. NACA 64A-010 airfoil, 899 X 161 nodes (143,680 cells) C-type mesh generated using Pointwise.	40
3.6	NACA 64A-010 airfoil with AFC jet (autonomously generated using the Pointwise glyph script).	41
3.7	Example of alternative meshing strategy utilized when $\theta \leq 15^\circ$ or $\theta \geq 165^\circ$	41
3.8	Comparison of turbulence models to experimental results of Smith and Walker [9] (baseline NACA 64A-010 airfoil).	43
3.9	Example of acceptable grid convergence during optimization.	43
4.1	Scaled design variables (top) and resulting objective function (bottom) for first 26 design points.	51

Figure	Page
4.2	Scaled design variables and resulting objective function for 46 iterations (final 20 generated using EGO). 52
4.3	20 best designs generated during optimization. 53
4.4	Trailing edge of design 9 ($Re = 0.95 \times 10^6$, $M = 0.75$, $\alpha = 0^\circ$, FUN3D, SA turbulence model). 55
4.5	Visualization of flow about design 9 compared to the baseline airfoil and aileron ($Re = 0.95 \times 10^6$, $M = 0.75$, and $\alpha = 0^\circ$). 56
4.6	Results of input uncertainty from variable Mach number, Reynolds number, and angle of attack. 57
4.7	Variable performance (ΔC_l) caused by increasing Mach number for two discrete angles of attack. 58
4.8	Flow visualization comparison of Design 9 for $\alpha = 0^\circ$ and $\alpha = 3^\circ$ ($M = 0.7884$, $Re = 0.95 \times 10^6$, FUN3D, SA turbulence model). 59
4.9	Ensemble of EDF's showing effects of input uncertainty including $\alpha = 1.5^\circ$ 60
4.10	Expanded ensemble of EDFs due to input uncertainty at $\alpha = -1^\circ$ and $\alpha = 4^\circ$ 61
4.11	Time accurate convergence history compared to steady-state. 62
4.12	Input uncertainty EDF's expanded to include model form uncertainty. 63
4.13	P-box including model form uncertainty and input uncertainty. 63
4.14	Grid convergence history (acceptable during optimization) for design 9 under mean flow conditions ($Re = 1.955 \times 10^6$, $M_\infty = 0.75$, and $\alpha = 1.5^\circ$). 64
4.15	Fully converged convergence history for design 9 under mean flow conditions ($Re = 1.955 \times 10^6$, $M_\infty = 0.75$, and $\alpha = 1.5^\circ$). 65

Figure		Page
4.16	Systematic mesh refinement study under mean flow conditions ($Re = 1.955 * 10^6$, $M_\infty = 0.75$, and $\alpha = 1.5^\circ$).	66
4.17	View of AFC jet (design 9) with coarsest mesh and Mach gradient.	67
4.18	P-box representing total predictive uncertainty.	68

List of Tables

Table		Page
3.1	The 20 cases evaluated for each design point to ensure a robust solution.....	31
4.1	Initial 20 designs specified by Dakota.	47
4.2	Initial 20 designs with constraint penalty.	49

List of Acronyms

AFC	Active flow control
CFD	Computational fluid dynamics
EDF	Empirical distribution function
EGO	Efficient global optimization
LHS	Latin hypercube sampling
SST	Menter’s shear stress transport turbulence model
NACA	National Advisory Committee for Aeronautics
NASA	National Aeronautics and Space Administration
OUU	Optimization Under Uncertainty
P-box	Probability-box
SA	Spalart-Allmaras turbulence model
SOGA	Single objective genetic algorithm
SRQ	System response quantity

List of Symbols

α	Angle of attack	N	Number of samples
b	Wing span	p	Roll rate
c	Chord length	$\dot{p}$	Time rate of change of roll rate
C_l	Coefficient of lift of airfoil	$\hat{p}$	Observed order of accuracy
C_p	Specific heat	p_f	Formal order of accuracy
C_{l_α}	Airfoil lift curve slope	q	Dynamic pressure
C_{L_p}	Angle of attack	r	Refinement factor
$C_{\dot{m}}$	Roll damping coefficient	Re	Reynolds number
d	Area validation metric	Re/c	Reynolds number per unit chord
d_{jet}	Diameter of base of AFC jet	ρ	Density
D_{jet}	Diameter of mouth of AFC jet	S	Wing area
δ_{turb}	Thickness of turbulent boundary layer	s	Arclength of the airfoil
Δ_1	Height of the cell nearest to the wall	$\sigma_{\Delta C_l}$	Sample standard deviation of change in lift
Δ_{C_l}	Change in airfoil lift coefficient	t	Time
$\overline{\Delta C_l}$	Mean change in lift	T_∞	Freestream temperature
ϵ	Discretization error	$T_{t,jet}$	Stagnation temperature at AFC jet
f	Updated objective function	u	x -component of velocity
F_s	Factor of safety	U_{DE}	Uncertainty due to discretization error
I_x	Moment of inertia about the x -axis	U_{IT}	Uncertainty due to iterative error
l	Lift generated by the airfoil	U_{NUM}	Uncertainty due to numerical error
ℓ	Location of center of AFC jet	U_{RO}	Uncertainty due to round-off error
L_p	Roll moment directional derivative	V	Velocity vector
λ	thermal conductivity	v	y -component of velocity vector
M	Mach number	y^+	Non-dimensional distance from wall
M_{crit}	Critical Mach number		
M_{DD}	Drag divergent Mach number		
$\mathcal{M}_{jet}$	Rolling moment generated by AFC jet		
$\dot{m}$	Mass flow rate through jet		
μ	Dynamic viscosity		

COMPUTATIONAL OPTIMIZATION UNDER UNCERTAINTY OF AN ACTIVE FLOW CONTROL JET

I. Introduction

1.1 Motivation and Proposed Solution

Modern military aircraft have smooth and deliberate geometry with limited sharp edges to limit observability and increase survivability [1]. However, that shape is not maintained throughout flight. Control surfaces, such as ailerons and rudders, change the geometric profile of the aircraft in order to manipulate the air around it and impart a change in force. This is effective for control purposes but requires a change in the carefully designed shape of the aircraft. As the shape changes, so too does the radar cross section (RCS). According to Zikidis, Skondras and Tokas, “Any irregularity of the surface could incur an RCS increase” [1]. The deflection of a control surface and the gap associated with that deflection are examples of surface irregularities. If it were possible to control an aircraft without changing its profile, the aircraft would theoretically maintain low observability throughout maneuvers.

A potential method for maneuvering without changing the shape of the aircraft is active flow control (AFC). One goal of this thesis is to evaluate the effectiveness of AFC to control an aircraft in transonic flight while maintaining a nearly constant shape. Optimization under uncertainty is performed to determine the most feasible implementation of a steady, blowing AFC jet that offers adequate control to an aircraft flying in the transonic regime. If deemed feasible, this AFC control surface could provide improved survivability to next generation aerial tankers, cargo air transport,

and transonic bombers.

Although increased survivability is the primary driver of this analysis, there are other potential advantages to the use of AFC for aircraft stability and control purposes. Traditional control surfaces account for a large percentage of the volume of an aircraft's wing, tail, and rudder. Ailerons and flaps, for example, typically take up about 15-25% of the wing chord. As a result of the limited volume available in the wing, particularly at the trailing edge, control surfaces are forced to be mechanically complex and structurally robust. A delicate balance is required between maintaining low weight, low volume, and the structural integrity required to resist the large hinge moments generated during controlled maneuvers. Mechanical complexity also increases manufacturing, assembly, and maintenance costs. [2]

It is possible that AFC could offer decreased complexity, weight, and volume. However, given the limited development of AFC actuators for stability and control when compared to traditional control surfaces, it cannot be guaranteed that the use of AFC would in actuality offer a decrease in complexity, weight, or volume. It is possible that the pumps, ducting, and valves required to achieve the design goals will increase the weight and complexity of the system or diminish power available for performance. By minimizing engine bleed, the design that is notionally most affordable to implement is determined.

1.2 Solution Approach

To effectively replace a traditional control surface, an AFC jet needs to be capable of imparting an equivalent or greater turning moment during any stage of flight. If the blowing or suction jet generates effective control authority at one speed but little to none at another speed, that jet could not replace a traditional control surface. For this reason, each design generated during optimization will be computationally

evaluated over a range of Mach numbers, Reynolds numbers, and angles of attack that an aircraft would expect to encounter during flight. The performance affected under each combination of uncertain flow conditions is included in optimization. This is known as design optimization under uncertainty.

An aircraft imparts control by changing the lift on one or more of its flight control surfaces. The AFC jet must, therefore, be capable of imparting a comparable or greater change in lift to a flight control surface. To ensure this, a constraint will be imposed that requires the design to generate an equivalent or better than prescribed change in lift. The mean change in lift and standard deviation over the range of uncertain inputs are each included in the constraints.

The mass flow rate required by the jet is minimized. Therefore, the design that requires the least amount of internal ducting and engine bleed is determined.

The four jet parameters, or design variables, that will be optimized are the total temperature ratio at the jet inlet, the angle of blowing, the location of the jet along the surface of the airfoil, and the width of the jet. These four design variables will be varied by the optimization algorithm until a near optimal combination is reached. The final improved solution will be evaluated to determine ease of implementation and whether the concept is worthy of further evaluation, such as 3-dimensional analysis and wind tunnel testing.

Uncertainty quantification is performed on the final improved design. Using the framework provided by Oberkampf and Roy [3], input uncertainty, model form uncertainty, and numerical uncertainty are estimated. A probability box is generated to summarize the total predictive uncertainty.

1.2.1 Scope

Rather than evaluating the entire flight regime, this study will focus on imparting control during cruise conditions, where low observability is more likely required. It is also more likely that the proposed AFC control surfaces will effectively replace the ailerons, rudder, or elevators used for control during cruise rather than the much larger flaps used during take-off and landing.

Rather than evaluate an entire aircraft or one of its wings, this thesis will evaluate an airfoil. If a blowing jet can generate a significant change in lift for an airfoil, one can reasonably assume that a similar change in lift could be generated by a three-dimensional wing, causing an aircraft to roll. Further analysis would be required to prove this assumption; however, only a two-dimensional airfoil will be analyzed over the course of this thesis.

In reality, each design is also subject to a plethora of uncertain inputs. However, only three variables are treated as uncertain during optimization to limit the number of computational evaluations. Additionally, only a select number of samples will be evaluated for each uncertain variable.

Similarly, only four design variables are allowed to vary during optimization. The design space is condensed by neglecting suction, unsteady forcing, and other potential AFC design considerations.

This study does not include empirical validation of the model. The analysis is purely computational. Experimental results are not readily available for comparison because the designs being analyzed are original. Baseline computational results without the AFC jet are used to estimate model form uncertainty.

1.3 Outcome and Impact

This analysis will result in an interesting and potentially valuable result with regards to the implementation of an AFC control surface for use in transonic cruise. Perhaps more importantly though, this analysis will provide a unique framework for optimization under uncertainty of an AFC jet on an airfoil. This analysis takes a stochastic optimization approach that incorporates into the constraints the mean and standard deviation of a sample flight regime. In doing so, the final solution is more likely to be statistically robust and effective over a range of flight conditions. Although this analysis is evaluating cruise conditions, it could be translated to evaluate climb, loiter, take-off, landing, or even multiple simultaneous flight regimes. This approach could also be extended to a specific aircraft by evaluating the design at statistically relevant conditions. Flight data associated with an operational aircraft would only further decrease the uncertainty of the proposed optimization approach.

1.4 Report Structure

Chapter 2 presents background research and theory regarding many of the fields associated with this analysis. Previous approaches to similar problems are also outlined. Chapter 3 includes a more detailed methodology and further clarification of the process and tools used for analysis. Chapter 4 provides results and detailed analysis. Chapter 5 specifies conclusions and recommendations for future research.

II. Background

2.1 Chapter Overview

This chapter provides a background and review of the literature for the topics covered throughout this thesis. Key concepts include active flow control (AFC), computational fluid dynamics (CFD), optimization under uncertainty (OUU), verification and validation, and uncertainty quantification.

2.2 Active Flow Control

Flow control can be broken into two broad categories, active and passive. Active flow control utilizes energy that is external to the system of interest to manipulate the flow, while passive flow control uses only the energy available within the control volume. A common example of passive flow control is a vortex generator. A vortex generator is purely geometric and does not require external energy to operate. However useful they may be, a vortex generator and all other passive flow control techniques are always in use, even when that use results in undesirable performance.

AFC, on the other hand, has the potential to be used intermittently or used in different manners during various stages of flight. The concept of AFC has been investigated for nearly 90 years [4]. As a result of the requirements for additional systems and power, however, the use of AFC has been largely limited to research and specialized aircraft. Limited high quality wind tunnel data has also contributed to the limited implementation of AFC through the years. According to Radaspiel et al. AFC experiments are costly and have historically resulted in error-prone data [4].

AFC encompasses a broad range of techniques. Three common AFC approaches are fluidic actuators, moving objects or surface actuators, and plasma actuators [5]. Fluidic actuators inject fluid to (blowing) or ingest fluid from (suction) the flow.

Moving object or surfaces utilize a moving surface to manipulate local fluid motion [5]. A morphing surface, such as the NASA morphing wing, is an example of a moving object or surface AFC. Plasma actuators produce steady jets by ionizing air from the freestream. Plasma actuators do not require moving parts and, therefore, require only an electric current and some form of conduit to operate [6]. Each AFC approach can be further expanded to a number of more specialized domains. Figure 2.1 shows a breakdown of a number of AFC actuators and techniques that have been investigated [5].

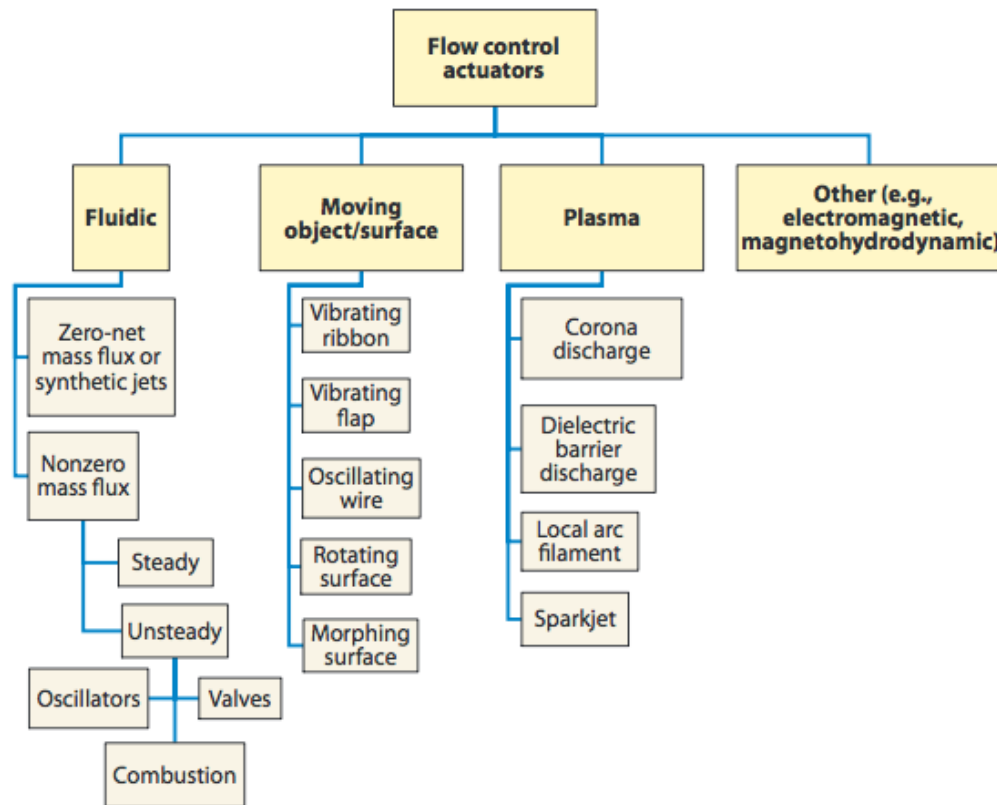


Figure 2.1. Classification of AFC actuators (reproduced from Ref.[5]).

This analysis will consider only fluidic actuators, particularly steady nonzero mass flux actuators. Steady blowing is particularly interesting as a result of its inherent simplicity. Steady blowing requires little to no moving parts especially when used

in conjunction with jet engines [4]. However, the plumbing and ducting required does induce a weight penalty. Plasma actuators also do not require moving parts, but steady fluidic actuators can generate higher velocity jets [5]. Although it has been shown that there are advantages to unsteady forcing [5], steady blowing or suction offers the option to perform simpler steady-state analysis. In addition to simpler computational analysis, experimental analysis is simpler for steady blowing or suction and is less prone to error. By limiting this evaluation to steady blowing, the problem is computationally simplified and theoretically more accurate.

The proposed solution will take advantage of steady blowing, but only for the short period of time in which a maneuver is performed. Unlike pure steady blowing, a control valve or similar device would be required. Given the requirement for a valve, unsteady actuation could be implemented in future studies.

2.3 AFC Implementation and Design Goals

Researchers have evaluated AFC to provide a number of different aerodynamic benefits. One proposed implementation of AFC is to delay separation at near-stall conditions. By delaying separation, an airfoil would remain effective at an angle of attack that was previously limited and would achieve a larger lift coefficient. With an increased lift coefficient, greater lift can be generated at lower speeds thereby enhancing performance during take-off and landing. With this goal in mind, Huang et al. performed an optimization study involving the implementation of an AFC steady jet on an NACA 0012 airfoil at a high angle of attack and low Reynolds number [7]. By varying jet location, amplitude, and angle, Huang et al. demonstrated that a steady jet could effectively increase the lift-to-drag ratio at 18° angle of attack using blowing, suction, and a combination of the two. They showed that suction created a large region of low pressure on the upper surface that increases lift, while keeping the

flow more attached to reduce pressure drag. Small amplitude downstream blowing also improved lift and drag characteristics, but by improving circulation.

Pehlivanoglu et al. took a similar approach to accomplish a different goal; increased aircraft performance in the transonic regime [8]. By varying the jet parameters and number of jets, Pehlivanoglu et al. demonstrated that a 20.7% improvement in lift-to-drag ratio is possible for a 2-dimensional NACA 64A-010 airfoil with three AFC jets. They attributed this improvement in part to the movement of the shock wave towards the trailing edge of the airfoil. Pehlivanoglu et al. validated computational results using the experimental results generated by Smith and Walker [9]. Their computational results showed "qualitative agreement" with the experimental data, but the difference was not quantified. Smith and Walker's experimental data is also used for validation by Qin et al. in a similar transonic study [10]. Due to its wide use and extensive results, this study also uses the experimental results of Smith and Walker for comparison.

Radaspiel et al. compared two blowing strategies: the Coanda-type wall jet and the vortex generator jet [4]. Like a traditional vortex generator, they showed that the vortex generator jet delayed flow separation and airfoil stall. The Coanda-type wall jet utilizes tangential blowing to provide increased lift by improving large flow turning on airfoils. Their analysis, like that of Huang et al. [7], focused on a low Reynolds number application typically associated with take-off and landing.

In a study similar to this one, Seifert et al. utilized AFC to impart roll control [11]. For low subsonic flight, they effectively demonstrated roll control of a small radio-controlled unmanned aircraft in both wind tunnel experiments and flight test. By activating an array of zero mass flux jets on a portion of one wing, they increased lift and executed a controlled roll. The work of Seifert et al. offers some proof of concept for the current study, but for very different flow conditions.

2.4 Computational Fluid Dynamics

This study relies on computational fluid dynamics (CFD) for analysis. CFD combines fluid dynamics, computer science, and mathematics to approximate a solution to the equations of fluid motion. “CFD is fundamentally based on the governing equations of fluid dynamics.” [12] The first governing equation is the continuity equation and states that mass cannot be created or destroyed. It is

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho V) = 0 \quad (2.1)$$

where ρ is density, t is time, and V is velocity vector of the flow. [12]

The second governing equation is the momentum equation, which stems from Newton’s second law of motion. In fluid flow, the conservation of momentum equations are known as the Navier-Stokes equations. Assuming a fluid continuum, an isotropic fluid, and negligible body forces, in two dimensions, the equations are

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} = -\frac{1}{\rho} \frac{\partial p}{\partial x} + \frac{\mu}{\rho} \frac{\partial^2 u}{\partial x^2} + \frac{\mu}{\rho} \frac{\partial^2 v}{\partial y^2} \quad (2.2)$$

$$\frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} = -\frac{1}{\rho} \frac{\partial p}{\partial y} + \frac{\mu}{\rho} \frac{\partial^2 v}{\partial x^2} + \frac{\mu}{\rho} \frac{\partial^2 v}{\partial y^2} \quad (2.3)$$

where x and y are Cartesian coordinates, u and v are the corresponding velocity components, and μ is the dynamic viscosity of the fluid.[12]

The final governing equation is the energy equation. It is derived from the first law of thermodynamics and can be mathematically expressed, in two dimensions, as

$$\frac{\partial T}{\partial t} + u \frac{\partial T}{\partial x} + v \frac{\partial T}{\partial y} = \frac{\lambda}{\rho C_p} \frac{\partial^2 T}{\partial x^2} + \frac{\lambda}{\rho C_p} \frac{\partial^2 T}{\partial y^2} \quad (2.4)$$

where λ is the thermal conductivity, T is the temperature, and C_p is the specific heat. [12]

The governing equations cannot be solved algebraically. Instead, CFD programs approximate a solution to a discretized form of the governing equations by iteratively solving the equations at a number of points within a flow field. The accuracy of the CFD solution approximation is dependent on the order of the discretization method and the resolution of the points in the flow field. The user must carefully specify boundary conditions and the points to be evaluated for CFD programs to reach an accurate solution. Before grid convergence, more points result in a more accurate solution, particularly in complex regions, but require greater computational expense.

CFD also takes into account turbulence, or random fluctuations in the fluid flow [12]. Nearly all fluid flows of engineering interest are turbulent. There are a number of turbulence models that work to approximate the effects of turbulence, each of which has unique advantages and disadvantages. The user must carefully select a turbulence model that is appropriate for the flow conditions and physical geometry of interest. Refer to Wilcox for additional detail about turbulence models [13].

2.5 Design Optimization

Design optimization utilizes mathematics to determine an optimum solution that meets a number of design requirements or constraints. In iterative solution methods, where the objective function may consist of nonlinear, coupled, partial differential equations (such as the Navier-Stokes equations) that may contain a discontinuous solution space, each new design is selected using “optimization concepts and procedures” which take into account previous designs [14]. The objective function must provide a numerical representation of a design’s merit. The objective function will be maximized or minimized depending on the design goals. Additionally, perfor-

mance requirements or design variable restrictions must be mathematically expressed as constraints.

There are a number of optimization strategies available. Each strategy differs in how it chooses the next design point. Some algorithms guarantee convergence to an optimum solution while others do not. Some algorithms provide a global search of the design space while others perform local searches. Each algorithm offers its own set of advantages and disadvantages.

Previous AFC optimization studies have used evolutionary algorithms [7] [8]. Evolutionary algorithms (also known as genetic algorithms) attempt to mimic Darwin’s theories of evolution and survival of the fittest. As is true in nature, the best designs are assigned a greater likelihood of reproducing and passing on their positive attributes to the next generation. To avoid premature loss of valuable traits or performance characteristics, genetic algorithms also incorporate the concept of mutation. Mutation diminishes the likelihood of the solution converging to a local optimum. As the algorithm advances, each generation produces an improved or comparable design until the global optimum solution is reached [14]. Although effective and global, evolutionary algorithms often require thousands or tens of thousands of function evaluations to reach an optimal solution [15]. To lessen the computational burden and reduce the number of expensive function evaluations for this study, a surrogate-based global optimization algorithm is utilized in this study to limit function evaluations.

Sandia National Laboratories’ Dakota surrogate-based global optimization algorithm generates a surrogate model that generally represents the first-order results of a more expensive computational model [15]. Using a Gaussian process, a response surface, or surrogate, is generated from a small sample of computational results. One such Gaussian process, known as Kriging, produces “smooth surface fit models of the response values from a set of data points.” [15] Further clarification of Dakota’s

Gaussian process algorithm is included in the Dakota User Manual [15].

The results associated with an initial sample must be provided to the algorithm in order to generate a surrogate. There are a number of ways to generate the initial sample. In this study, the initial sample is generated randomly using Latin hypercube sampling (LHS). LHS guarantees diversity of initial designs. Rather than selecting random design points, the range of each uncertain variable is split into N bins, where N is the number of samples requested. For each uncertain variable, a sample is selected randomly from each bin [15]. The variables are randomly combined to form N initial designs that span the design space.

A surrogate can be optimized using a number of optimization techniques. In this study, the surrogate is optimized using an evolutionary algorithm, specifically SOGA (single objective genetic algorithm). An evolutionary algorithm is chosen for its global search capability. A global optimizer will return the optimal point from the entire design space rather than a local optimum. Evolutionary algorithms are also effective in discrete solution spaces. However, this advantage is not relevant in this study because Gaussian process surrogates are inherently continuous.

In surrogate-based optimization, the surrogate is optimized. The optimal surrogate design is evaluated using the high-fidelity computational solver, the surrogate is updated, and the process is repeated until the stopping criteria is met and a near-optimal solution is reached. This process takes advantage of the global search made possible by genetic algorithms while limiting the number of computationally expensive function evaluations required. The process is shown in Figure 2.2.

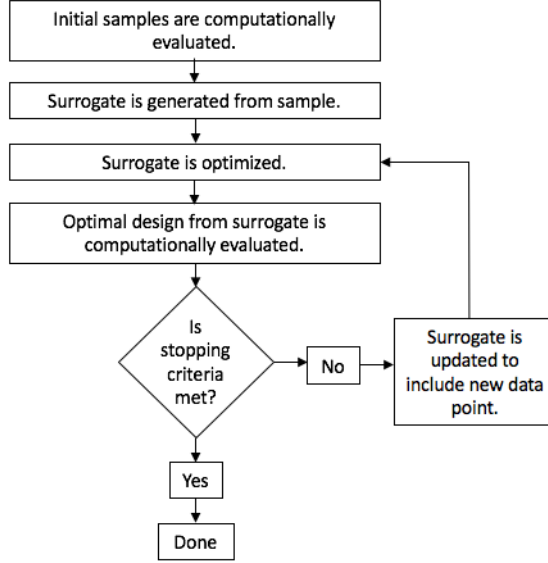


Figure 2.2. Surrogate-based global optimization.

The surrogate-based optimization strategy selects the predicted best design for computational analysis. Therefore, it exploits the design space but does not intentionally explore it. Another surrogate optimization technique, efficient global optimization (EGO), strikes a balance between exploration and exploitation. Therefore, it is more likely to efficiently improve a surrogate. EGO maximizes an expected improvement function. The expected improvement function incorporates statistics to predict what design point is most likely to produce an improved objective function. If a region of the design space is under-explored, the uncertainty in the objective is large and so is the potential for improvement. EGO generates the surrogate in the same way as surrogate-based global optimization (using a Gaussian process). It differs only in its selection of the next design point to evaluate. [16]

2.6 Optimization under Uncertainty (OUU)

OUU is a nondeterministic approach to determining the best solution. The inputs are treated as uncertain. For each uncertain input, a number of samples represen-

tative of the uncertain variable distribution are evaluated. For each combination of uncertain inputs, a unique response is generated; thus a given array of inputs yields a corresponding array of solutions. Statistics on the array of solutions are incorporated into the objective and constraint functions. [15]

Uncertain inputs can be classified as either aleatory or epistemic. Aleatoric uncertainty is inherently random and follows some known distribution. Epistemic uncertainty, on the other hand, reflects a lack of knowledge and does not follow a distribution or follows a distribution that is unknown or not well understood. [3]

Although OUU is not a new concept, this study offers the unique contribution of applying it to AFC. Freeman and Roy utilized OUU to optimize several parameters of 3-dimensional tractor-trailer base-drag reduction devices. By optimizing under uncertain wind speed, direction, and flap deflection angle, they determined a relatively robust design. In their study, wind speed and direction were treated as aleatory input variables and flap deflection angle was treated as epistemic. They also applied additional sources of uncertainty to the optimized solution to quantify total predictive uncertainty. The general process of this study will closely follow that of Freeman and Roy. [17]

2.7 Verification, Validation, and Uncertainty Quantification

In order to determine total predictive uncertainty of a given model, as is done in this study, a verification and validation study must be completed. Verification is the process of ensuring that the equations being solved in computations are being solved properly. Validation is the process of ensuring that the equations being solved in computations are appropriately applied to the problem. In other words, validation ensures that the right equations are being solved and verification ensures that the equations are being solved right [3].

2.7.1 Verification and Numerical Uncertainty

Verification can be broken into two distinct categories, code verification and solution verification. Code verification is the practice of determining that a specified code is properly implemented. In scientific computing, the exact solution is often not available. In these cases, the solution can not be checked for correctness in the preferred manner. Instead codes can be verified in a number of different ways. In order of increasing rigor, these include visual checks, code-to-code comparisons, discretization error quantification, convergence tests, and order of accuracy tests. The order of accuracy test, the most rigorous code verification practice, is the comparison of the formal order of accuracy with the observed order of accuracy. Order of accuracy is the rate at which discretization error decreases with systematic mesh refinement. The observed order of accuracy must asymptotically approach the formal order of accuracy as the mesh is systematically refined. The formal order of accuracy is the theoretical rate at which the discretization error decreases while the observed order of accuracy is the rate at which the discretization error actually decreases with systematic mesh refinement. The observed order of accuracy ($\hat{p}$) can be determined by comparing the results of three systematically refined meshes using

$$\hat{p} = \frac{\ln\left(\frac{f_3 - f_2}{f_2 - f_1}\right)}{\ln(r)} \quad (2.5)$$

where f_3 is the solution determined using the coarsest of three systematically refined meshes, f_2 is the solution determined using the second finest of three systematically refined meshes, f_1 is the solution determined using the finest of three systematically refined meshes, and r is the refinement factor. [3]

Systematic mesh refinement is the uniform and consistent refining of grids or meshes. Uniform refinement is the practice of refining grids by the same factor in all

directions over the entire grid. Consistent refinement retains or improves the mesh growth rate, curvature, skewness, and aspect ratio. [3]

Solution verification is the practice of estimating numerical error. Discretization error, round-off error, and iterative error each contribute to total uncertainty due to numerical error. The equation for total uncertainty due to numerical error (U_{NUM}) is

$$U_{NUM} = U_{DE} + U_{IT} + U_{RO} \quad (2.6)$$

where U_{DE} is the uncertainty due to discretization error, U_{IT} is the uncertainty due to iterative error, and U_{RO} is the uncertainty due to round-off error. [3]

Iterative error is the difference between the current solution and the fully iterated solution. If the solution is allowed to run until the residuals have decreased by at least seven orders of magnitude, the iterative error is assumed to be zero. Otherwise, it can be estimated by comparing a solution with a finite number of iterations to a fully converged solution in which the residuals have decreased by at least seven orders of magnitude. To provide an accurate estimate of iterative error, one of the simulations is allowed to converge entirely.

Round-off error is the result of using finite arithmetic on digital computers [3]. Computers use a finite number of digits in every computation. This can lead to error as rounded numbers are multiplied by one another over time. Using double-precision, round-off error is typically very small and is approximated as one percent of the discretization error in this study.

Discretization error is the difference between the exact solution to the discretized equations and the exact solution to the governing differential equations [3]. Because the exact solution to the governing equations is often unknown, discretization error can be estimated using Generalized Richardson Extrapolation which estimates

discretization error (ϵ_i) according to

$$\epsilon_i = \frac{f_{i+1} - f_i}{r^p - 1} \quad (2.7)$$

where i is representative of the mesh being evaluated, $i + 1$ is the mesh one level coarser, and p is the order of accuracy of the discretized equations. The uncertainty due to discretization error, U_{DE} , is estimated using

$$U_{DE} = F_s * \epsilon_i \quad (2.8)$$

where F_s is a factor of safety, which can be varied depending on the reliability of the solutions.

The reliability of solutions can be determined by comparing the observed order of accuracy over a series of systematically refined meshes to the formal order of accuracy, p_f . If the observed order of accuracy falls within 10% of the formal order of accuracy, the solution can be considered reliable and a factor of safety of 1.25 can be used. Otherwise, a factor of safety of 3 should be used to insure that the uncertainty due to discretization error is not underestimated [3].

2.7.2 Validation and Model Form Uncertainty

A common validation approach in engineering is the comparison of computational results to experimental results. The percent difference between experimental and computational results is often provided to quantify this validation. A more rigorous approach known as the area validation metric can also be used, provided a number of design points are available for comparison.

The area validation metric (d) is the area between two empirical distribution functions (EDF), one representing experimental results and the other representing

computational results. An EDF is an experimentally generated estimation of the cumulative distribution function that has one or more uncertain inputs. An EDF is created by computing the system response quantity (SRQ) for each combination of aleatoric inputs for a fixed value of the epistemic input. [3]

The area validation metric is defined algebraically as

$$d = \int_{-\infty}^{\infty} |F(Y) - S_n(Y)| dY \quad (2.9)$$

where Y is the SRQ, $F(Y)$ is the computational EDF, and $S_n(Y)$ is the experimental EDF. The area validation metric can be used to estimate model form uncertainty by comparing a number of area validation metrics to one another. Model form uncertainty results from the differences between the model and the real-world event that the model is attempting to capture. An example of the area validation metric between two distinct EDF's is included in Figure 2.3 [18]. Each EDF represents a constant epistemic variable with variable aleatory inputs. The EDF from the simulation looks smooth because a large number of samples, N , were computationally evaluated. The EDF from the experiment is blocky because only four experimental samples are available for comparison.

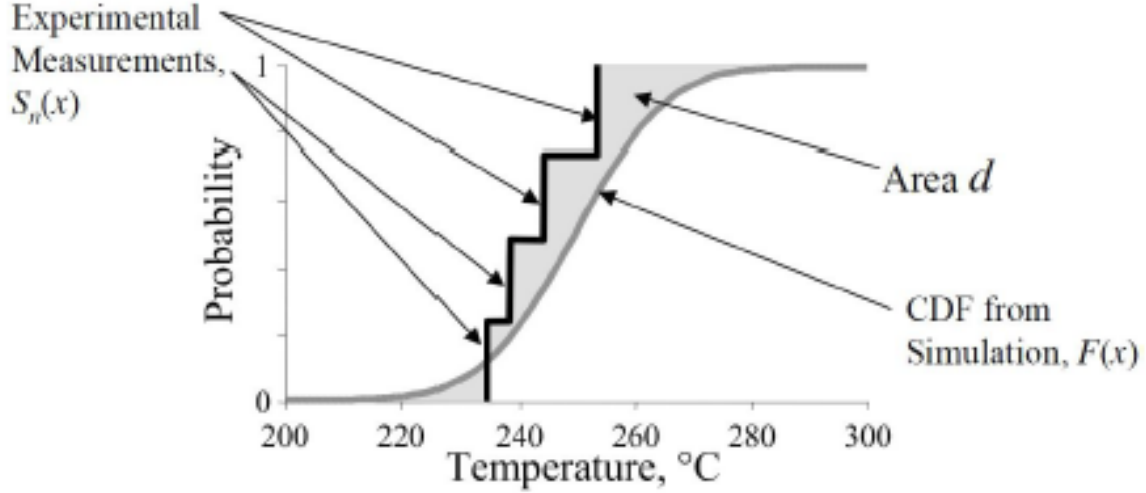


Figure 2.3. Example area validation metric between two distinct EDF's. (reproduced from Ref. [18])

In this study experimental results are not available for comparison because the designs being analyzed are original. Different turbulence models, computational solvers, boundary conditions, and time steps each predict different results to the same real-world event. Without an adequate validation experiment, it is not possible to say which prediction is most accurate. Therefore, the model form uncertainty is less understood and shaped by the differences between results. In such cases, the maximum difference between results serves as a conservative estimate of the model form uncertainty.

2.7.3 Total Predictive Uncertainty

Total predictive uncertainty is a combination of numerical uncertainty, input uncertainty, and model form uncertainty. Input uncertainty is the range of solutions falling within the bounds of all EDF's associated with a particular solution. There will be an EDF for each epistemic input that is evaluated. The input uncertainty includes the uncertainty associated with the aleatory and epistemic variables because each EDF takes into account aleatoric variability.

A probability box (p-box) is a visual representation of total predictive uncertainty. It allows the reader to quickly differentiate between sources of uncertainty. The center of the p-box is the input uncertainty and encompasses all of the EDF's. The second tier is the model form uncertainty with a width of d , applied to both upper and lower extents of the input uncertainty. The third tier, the outside portion, is the numerical uncertainty. For clarification, an example p-box representing total predictive uncertainty is included in Figure 2.4 [19]. In the example, the SRQ is thrust.

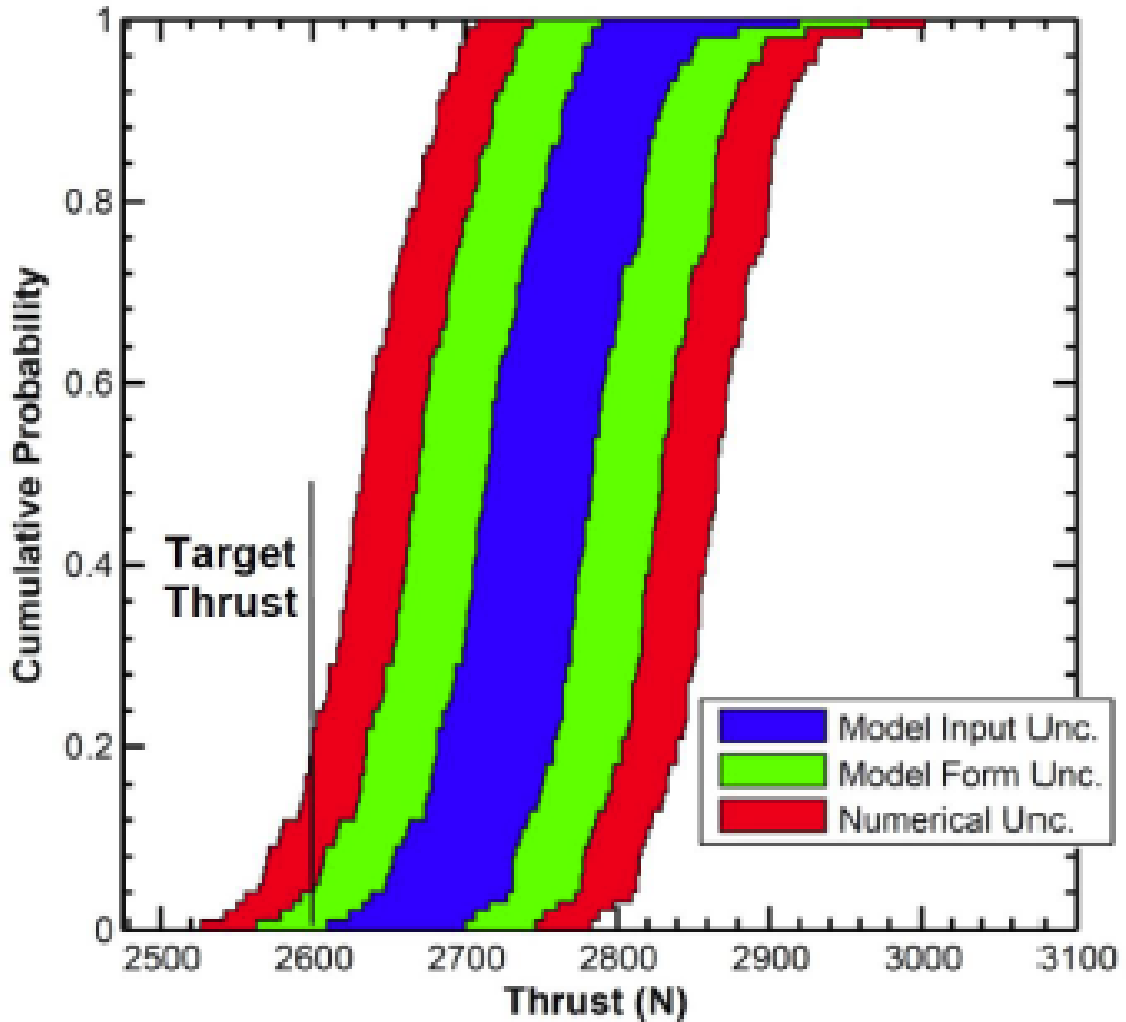


Figure 2.4. Example p-box representing total predictive uncertainty. (reproduced from Ref. [19])

2.8 Summary

AFC jets have not been optimized to impart control during transonic flight. Further, OUU has yet to be applied to the AFC jet problem. This study addresses each of these tasks utilizing theory and previous work laid forth in this chapter. Uncertainty will also be quantified for the final design.

III. Research Methodology

3.1 Chapter Overview

A single blowing AFC jet is optimized under uncertainty to impart control to an airfoil during transonic cruise conditions. The nondimensional mass flow rate through the jet is minimized. The jet location, angle of blowing, total temperature ratio at the inlet, and jet diameter are variable. Optimization is performed using Dakota's surrogate-based global optimization algorithm. Each design is prescribed by Dakota, then computational model geometry is generated using Pointwise. Each design is evaluated using FUN3D under 20 unique combinations of Mach number, Reynolds number, and angle of attack. The mean change in lift and standard deviation are determined for each design and returned to Dakota as constraints. The next design point is prescribed by Dakota and the process repeats. The optimization process continues until there is little to no improvement in the mass flow rate between iterations. The process is graphically expressed in Figure 3.1.

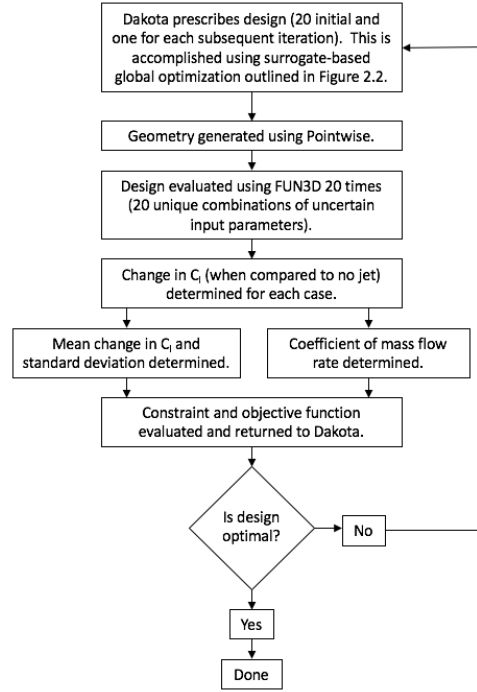


Figure 3.1. Graphical representation of optimization process.

3.2 Sampling Approach

To effectively replace a standard control surface, the AFC jet must be capable of generating control over the entire flight regime. To insure this, each design is evaluated over a range of uncertain flow conditions representative of transonic cruise.

An airliner or cargo jet cruises at a number of varying flight conditions. Its altitude and speed vary over the course of each flight. So too do the characteristics of the air through which it is flying. The temperature, density, and wind direction are constantly changing. The weight of an aircraft changes as fuel is burned and from flight to flight. The number of cruise conditions an aircraft could encounter is nearly infinite. A finite number of flight conditions, therefore, must be carefully selected.

3.2.1 Mach Number

Transonic flight is the region of flight during which some portion of the flow exceeds the speed of sound. Aircraft tend to cruise in the transonic regime at a velocity that allows for the greatest balance between fuel efficiency and timeliness. Drag increases substantially in the transonic regime due to the formation of shock waves and the onset of wave drag. This is why the sound barrier was widely viewed as unbreakable in the early years of aviation. As an aircraft approaches the speed of sound, the drag increases substantially. The Mach number at which this sharp rise in drag occurs is called the drag divergence Mach number (M_{DD}). [2]

The exact value of the M_{DD} is somewhat arbitrary because the drag rise is not instantaneous. As an example, a typical drag coefficient versus Mach number plot is included in Figure 3.2 [20]. In order to estimate M_{DD} , many of the large aerospace companies have adopted their own definition. The Boeing Company defines M_{DD} as the Mach number at which the drag coefficient increases by 0.0020. This usually corresponds to a Mach number about 0.08 greater than the critical Mach number ($M_{DD_{Boeing}} \approx M_{crit} + 0.08$). The critical Mach number (M_{crit}) occurs where supersonic flow first exists at some point over the airfoil. On the other hand, The Douglas Company defined M_{DD} as the Mach number at which the rate of change of drag

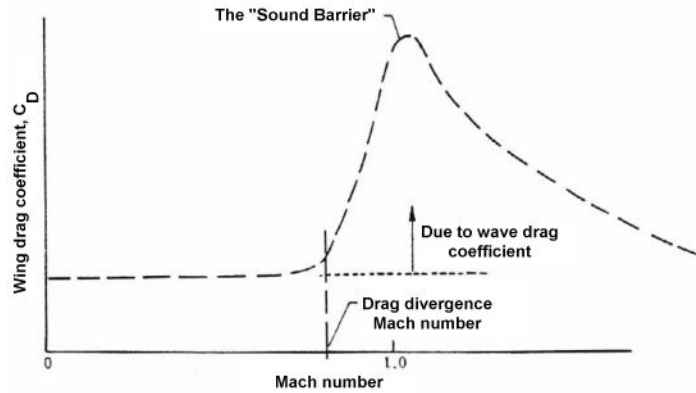


Figure 3.2. Drag rise due to wave drag in the transonic flight regime. (reproduced from Ref. [20])

with respect to Mach number first reaches 0.10. This corresponds to a larger Mach number than that of Boeing, typically about 0.14 greater than M_{crit} ($M_{DD_{Douglas}} \approx M_{crit} + 0.14$). According to Raymer, jet transports usually cruise at Boeing’s definition of M_{DD} and have a maximum level speed about equal to Douglas’s definition. These estimations assume an airfoil that is not supercritical. [2]

Assuming Raymer’s estimate is accurate, the cruise Mach number of a transport jet is approximately equal to

$$M_{cruise} \approx M_{DD_{Boeing}} \approx M_{crit} + 0.08 \quad (3.1)$$

This cruise Mach number, however, is variable. Winds are constantly changing over the course of a flight causing minor changes in airspeed. A pilot may also choose to fly at an airspeed less than or greater than what is typically recommended for various reasons such as air traffic, timeliness, or mission requirements. Although the true velocity variation is unknown (epistemic), this study assumes a normal distribution about $M_{DD_{Boeing}}$ (aleatoric). This assumption allows for the development of EDF’s and a p-box to quantify input uncertainty. The $M_{DD_{Douglas}}$ is treated as two standard deviations greater than the mean cruise velocity. This assumption ensures that there is only a 2.27% probability of exceeding what Raymer describes as the maximum level speed.

Cruise Mach is treated as the freestream Mach number of the air at the boundary of the grid being investigated. Therefore, it does not matter whether the aircraft is flying into a steady head wind or with a steady tail wind. Sideslip is not considered due to the 2-dimensional nature of this analysis. And by evaluating the airfoil at speeds relative to its own critical Mach number, this analysis is theoretically independent of airfoil geometry.

A modified Latin hypercube sampling approach is used to sample from the as-

sumed normal distribution. Each design point is evaluated over five Mach numbers in an attempt to represent the normal distribution while limiting computational requirements. Rather than selecting a random Mach number from each bin, the five selected Mach numbers will come from the center of each of the five bins. This is equivalent to sampling at cumulative probabilities of 0.1, 0.3, 0.5, 0.7, and 0.9. Although this limits the variability of analysis, it allows for a direct comparison between each design. The sampling approach used to select cruise Mach numbers is illustrated in Figure 3.3.

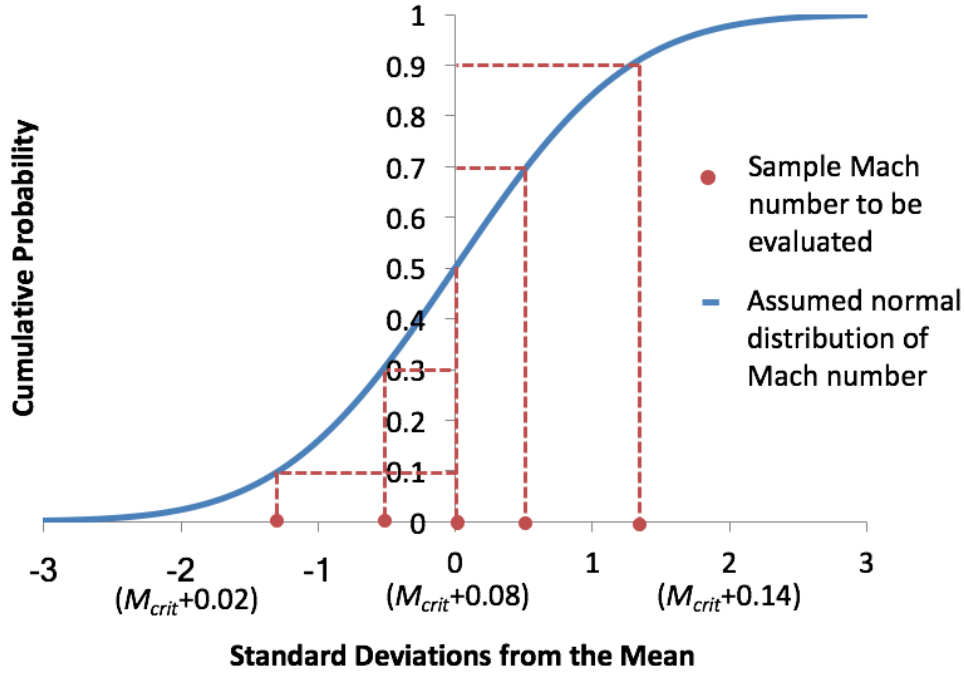


Figure 3.3. Sampling approach utilized to select cruise Mach numbers.

The critical Mach number is not constant during flight; it changes with angle of attack and Reynolds number. However, this study will assume a constant M_{crit} . To determine a representative M_{crit} , a number of Mach numbers were computationally analyzed under the presumed average flow conditions ($\alpha = 1.5^\circ$ and $Re = 1.955 \times 10^6$). The freestream Mach number at which supersonic flow is first achieved is,

by definition, the critical Mach number. At these average flow conditions, FUN3D returned M_{crit} of 0.69 for the the NACA 64A-010 airfoil.

3.2.2 Reynolds Number

The cruise altitude of a typical transonic aircraft is also subject to variability. The cruise altitude is influenced by the distance of the flight. Although there is an ideal cruising altitude for each aircraft, it may not be worth the fuel to climb to that altitude for a short flight. Storms, air traffic, and mission requirements also cause variability in cruising altitude. Altitude, therefore, does not necessarily follow a normal distribution and will be treated as an epistemic variable. The maximum and minimum cruise altitudes a transonic aircraft might experience are analyzed. Those altitudes are assumed to be 20,000 and 40,000 feet, respectively. The CFD solver, FUN3D, does not accept altitude as an input. Instead Reynolds number per unit chord length will be varied to represent changes in altitude. This also accounts for potential variations in atmospheric conditions, such as local fluctuations in density, temperature, and pressure. Reynolds number and Mach number are not independent but are each required to properly specify freestream conditions. Assuming standard atmospheric conditions and the mean cruise Mach number, the Reynolds numbers per unit chord length corresponding to 20,000 and 40,000 feet are 0.95 and 2.96 million, respectively. The equation used to determine Reynolds number per unit chord length (Re/c) of an airfoil is

$$Re/c = \frac{\rho_{\infty} u_{\infty}}{\mu_{\infty}} \quad (3.2)$$

where ρ_{∞} is the density of the freestream air, μ_{∞} is the kinematic viscosity of the freestream air, and u_{∞} is the x-component of velocity of the freestream air.

3.2.3 Angle of Attack

The angle of attack of the airfoil (α) is also variable during cruise. Even when flying straight and level, the angle of the wing relative to the freestream air could vary substantially. The angle of attack of the airfoil contributes to the lift generated by the airfoil (l). At angles of attack well beneath the angle of stall, as is the case in cruise, the lift generated by a 2-dimensional airfoil is

$$l = C_l q_\infty c \quad (3.3)$$

$$C_l = C_{l_\alpha} \alpha \quad (3.4)$$

where C_{l_α} is the lift-curve slope of the particular airfoil and q_∞ is the dynamic pressure of the freestream air [21].

In steady, level, unaccelerated flight, the angle of attack is set such that lift is equal to weight. Under constant freestream conditions, the angle of attack is the only factor which can change the lift. The angle of attack, therefore, must be adjusted to account for the weight of the aircraft in order to maintain steady level flight. If an aircraft is fully fueled and carrying its maximum payload, the angle of attack must be relatively large. If an aircraft is empty and low on fuel, the angle of attack is relatively small. The weight of an aircraft is also constantly changing as fuel is burned and in the case of munition or cargo drops, the weight can change substantially in an instant.

To limit fuselage drag during cruise, aircraft designers generally design the wings at a fixed angle of attack, or the angle of incidence, when the fuselage is level. The angle of incidence is chosen such that the fuselage is nearly level for the majority of cruise conditions. This insures minimum fuselage drag during cruise. “General aviation and home built aircraft have an incidence of about 2° , transport aircraft about 1° , and military aircraft approximately zero” [2].

This analysis focuses on improving survivability, possibly for use by military aircraft. However, the airfoil of interest is nearly symmetric and would likely be flown with an angle of attack greater than 0° to attain the lift required to maintain level cruise. Given the general nature of this analysis, it is not possible to determine the true distribution of the angle of attack. Using Raymer's estimates, however, this study predicts the range of angles over which the airfoil might be flown. To insure that the entire design space is evaluated, a maximum and minimum angle of attack of 3° and 0° will be evaluated, respectively.

3.2.4 Sampling Summary

Each design is evaluated at five Mach numbers, two Reynolds numbers, and two angles of attack. Thus, each design will be evaluated at 20 unique combinations of flow conditions. The design points to be evaluated are detailed in Table 3.1.

Case	Reynolds Number (*10 ⁶)	Angle of Attack (°)	Mach Number
1	0.95	0	M _{crit} +0.0216= 0.7116
2	0.95	3	M _{crit} +0.0216= 0.7116
3	0.95	0	M _{crit} +0.0441= 0.7341
4	0.95	3	M _{crit} +0.0441= 0.7341
5	0.95	0	M _{crit} +0.06= 0.75
6	0.95	3	M _{crit} +0.06= 0.75
7	0.95	0	M _{crit} +0.0759= 0.7659
8	0.95	3	M _{crit} +0.0759= 0.7659
9	0.95	0	M _{crit} +0.0984= 0.7884
10	0.95	3	M _{crit} +0.0984= 0.7884
11	2.96	0	M _{crit} +0.0216= 0.7116
12	2.96	3	M _{crit} +0.0216= 0.7116
13	2.96	0	M _{crit} +0.0441= 0.7341
14	2.96	3	M _{crit} +0.0441= 0.7341
15	2.96	0	M _{crit} +0.06= 0.75
16	2.96	3	M _{crit} +0.06= 0.75
17	2.96	0	M _{crit} +0.0759= 0.7659
18	2.96	3	M _{crit} +0.0759= 0.7659
19	2.96	0	M _{crit} +0.0984= 0.7884
20	2.96	3	M _{crit} +0.0984= 0.7884

Table 3.1. The 20 cases evaluated for each design point to ensure a robust solution.

3.3 Problem Formulation

3.3.1 Objective Function

The AFC jet is optimized to minimize the mass flow rate required to achieve a robust change in lift for an airfoil. By minimizing the mass flow rate, a design that achieves the performance objectives and requires the least amount of engine bleed and internal ducting is determined. In this analysis, the coefficient of mass flow rate $C_{\dot{m}}$ is used as the objective function. $C_{\dot{m}}$ is defined as [4]

$$C_{\dot{m}} = \frac{\dot{m}_{jet}}{\rho_{\infty} V_{\infty} S_{ref}} = \frac{\rho_{jet}}{\rho_{\infty}} \frac{d_{jet}}{c} \frac{|V_{jet}|}{|V_{\infty}|} \quad (3.5)$$

where ρ_{jet} is the density of the air leaving the jet, $|V_{jet}|$ is the velocity magnitude of the air leaving the jet, and d_{jet} is the diameter of the base of the jet.

3.3.2 Design Variables and Physical Constraints

The four design variables for the optimization include:

- ℓ/s : location of the center of the jet along the surface of the airfoil (*unitless*)
- D_{jet}/s : diameter of the jet on the surface of the airfoil (*unitless*)
- $T_{t,jet}/T_\infty$: ratio of total (stagnation) temperature at the jet to the freestream temperature (*unitless*)
- θ : angle of the flow exiting the jet relative to the surface of the airfoil (*degrees*)

where s is the arclength or the length of the skin of the airfoil. Each of the four design variables is allowed to vary to determine the optimal or near-optimal solution. The design variables are graphically expressed in Figure 3.4. Notice the difference between the design variable, D_{jet} and the dependent variable, d_{jet} (used to determine $C_{\dot{m}}$).

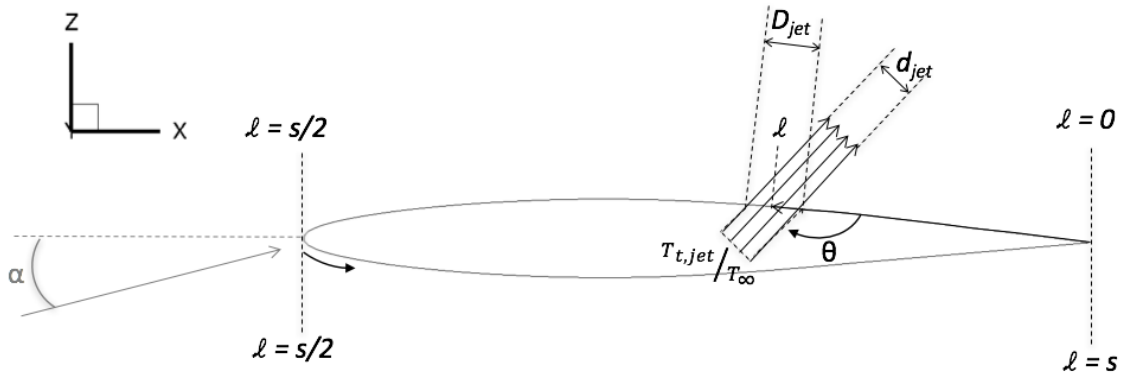


Figure 3.4. Graphical representation of each design variable (NACA 64A-010 airfoil).

The relationship between D_{jet} and d_{jet} is shown in Equation 3.6.

$$D_{jet} = \frac{d_{jet}}{\sin\theta} \quad (3.6)$$

D_{jet}/s is limited to a maximum value of 1.25% (approximately 2.5% of the chord). This constraint ensures that a jet can be realistically modeled as a single channel in the airfoil. Dannenberg and Weiberg showed that increasing the area of AFC to greater than 2.5% of the chord does not significantly increase lift [22]. The diameter must also be positive to remain physically possible.

The location of the jet is limited only by geometry. It is allowed to vary between 0 and the arclength, s (the length of the skin of the airfoil). $\ell/s = 0$ is the upper surface trailing edge, $\ell/s = 1/2$ is the leading edge, $0 < \ell/s < 1/2$ is the upper surface, $1/2 < \ell/s < 1$ is the lower surface of the airfoil, and $\ell/s = 1$ is the lower surface trailing edge. The following constraints are required to ensure that the jet does not extend past the trailing edge on the lower or upper surface.

$$\ell/s - \frac{D_{jet}/s}{2} \geq 0 \quad (3.7)$$

$$\ell/s + \frac{D_{jet}/s}{2} \leq 1 \quad (3.8)$$

The AFC jet is treated as a nozzle. At a nozzle, the magnitude of the flow is limited to Mach 1 due to mass flow choking [21]. Therefore, the total temperature ratio ($T_{t,jet}/T_\infty$) must be limited such that the flow does not exceed Mach 1. The jet exit velocity (V_{jet}) can be approximated as a function of $T_{t,jet}/T_\infty$ by assuming an isentropic process and $T_{jet} = T_\infty$. Under these assumption, the jet exit velocity is [23]

$$V_{jet} = \sqrt{\frac{2\gamma R T_{t,jet}}{\gamma - 1} \left(1 - \left(\frac{P_\infty}{P_{t,jet}}\right)^{\frac{\gamma-1}{\gamma}}\right)} \quad (3.9)$$

where γ is the ratio of specific heats and R is the gas constant for air. From the isentropic relations,

$$\frac{P_{t,jet}}{P_\infty} = \frac{P_{t,\infty}}{P_\infty} \frac{P_{t,jet}}{P_{t,\infty}} = \frac{P_{t,\infty}}{P_\infty} \left(\frac{T_{t,jet}/T_\infty}{T_{t,\infty}/T_\infty}\right)^{\frac{\gamma-1}{\gamma}} \quad (3.10)$$

And for the average freestream conditions evaluated ($M_\infty = 0.75$),

$$\frac{P_{t,jet}}{P_\infty} = 1.4523 \left(\frac{T_{t,jet}/T_\infty}{1.1125} \right)^{\frac{\gamma-1}{\gamma}} \quad (3.11)$$

To remain subsonic, V_{jet} cannot exceed the speed of sound (a). The speed of sound for an ideal gas is

$$a = \sqrt{\gamma R T} \quad (3.12)$$

The smallest speed of sound evaluated in this study occurs at 40,000 feet and is equal to 297.2 meters per second. To ensure that V_{jet} remains subsonic and positive, $T_{t,jet}/T_\infty$ is constrained between 0.302 and 1.581. These bounds are determined by solving Equations 3.9 and 3.11 for $V_{jet} = 0$ and $V_{jet} = 297.2$ m/s.

The angle of the AFC jet flow is limited between 3 and 177 degrees to preclude non-physical co-planar flow. This angle is measured clockwise from the surface in the direction of the interior, as shown in Figure 3.4.

3.3.3 Performance Constraint

During cruise, roll control is often utilized. Roll control is provided by changing the lift equally and opposite on each of the wings to impart a rolling moment about the longitudinal axis. An operator typically accomplishes this using ailerons, spoilers, or a combination of the two. The AFC jet must generate a comparable change in lift to adequately replace these control surfaces.

The European Aviation Safety Agency requires that cruising jet transport aircraft can perform a 60-degree roll in 11 seconds [24]. This serves as a baseline for adequate roll control. The Federal Aviation Administration does not specify a minimum rate of roll during cruise.

The roll rate of an aircraft (p) can be expressed as a differential equation. Equation

3.13 is derived from the summation of moments about an aircraft.

$$I_x \dot{p} = \mathcal{M}_{jet} - L_p I_x p \quad (3.13)$$

I_x is the moment of inertia about the x -axis (longitudinal axis), $\mathcal{M}_{jet}$ is the moment generated by the AFC jet, $\dot{p}$ is the time rate of change of the roll rate, and L_p is the roll moment directional derivative resulting from the aircraft's resistance to rolling. It is mathematically expressed as

$$L_p = \frac{q S b^2 C_{L_p}}{2 I_x V_\infty} \quad (3.14)$$

where S is the area of the wing, b is the wing span, and C_{L_p} is the roll damping coefficient due to wing surface area.

The moment required to perform a 60-degree roll in 11 seconds is determined by assuming an initial roll rate of zero. Using the aircraft data provided by Nelson [25], the moment required to roll a Boeing 747-200 is 77,120 pound-feet. The ailerons on a Boeing 747-200 encompass approximately 30% of the outboard portion of the wing with the exception of 5% at the wingtip [25]. If AFC jets were implemented over the same 25% of each wing, that portion of wing would require a change in airfoil lift coefficient (ΔC_l) of 0.1022. In contrast, a narrow-bodied Convair 880 jet transport would require a ΔC_l of just 0.0335 to accomplish the same task. Although a Boeing 747 is a larger aircraft than most cargo aircraft, the change in lift required to roll serves as the minimum acceptable change in lift in this analysis. The constraint is expressed in Equation 3.15 in standard form.

$$0.1022 - \overline{\Delta C_l} + 2\sigma_{\Delta C_l} \leq 0 \quad (3.15)$$

$\overline{\Delta C_l}$ is the mean change in lift of all of the cases evaluated and $\sigma_{\Delta C_l}$ is the sample

standard deviation. By organizing the constraint in this manner, each qualifying jet design has a 97.7% probability of meeting or exceeding a $\overline{\Delta C_l}$ of 0.1022 over the entire range of predicted cruise conditions (assuming a normal distribution). ΔC_l is determined for each computation by comparing the coefficient of lift generated by the airfoil with the AFC jet to the same airfoil without one. $\overline{\Delta C_l}$ and $\sigma_{\Delta C_l}$ are determined for each design point using Eqs. 3.16 and 3.17. Subscript i represents each of the 20 cases outlined in Table 3.1.

$$\overline{\Delta C_l} = \frac{1}{20} \sum_{i=1}^{20} \Delta C_{l_i} \quad (3.16)$$

$$\sigma_{\Delta C_l} = \frac{1}{19} \sum_{i=1}^{20} (\Delta C_{l_i} - \overline{\Delta C_l})^2 \quad (3.17)$$

3.3.4 Optimization Parameters

Dakota's surrogate based global algorithm is utilized for construction of the surrogate. Dakota's SOGA algorithm is used to optimize the surrogate. The recommended default parameters are utilized for each. The seed used by SOGA is 10983. The initial sample is generated using LHS and the seed, 531.

3.3.5 Optimization Problem

The optimization problem is summarized in Equation 3.18.

$$\begin{aligned}
&\text{Minimize} && C_{\dot{m}} = \frac{\rho_{jet}}{\rho_{\infty}} \frac{d_{jet}}{c} \frac{|V_{jet}|}{|V_{\infty}|} \\
&\text{subject to} && \overline{\Delta C_l} - 2\sigma_{\Delta C_l} \geq 0.1022 \\
&&& 0 \leq D_{jet}/s \leq 0.0125 \\
&&& 3^{\circ} \leq \theta < 177^{\circ} \\
&&& 0.302 \leq T_{t,jet}/T_{\infty} < 1.581 \\
&&& \ell/s - \frac{D_{jet}/s}{2} \geq 0 \\
&&& \ell/s + \frac{D_{jet}/s}{2} \leq 1
\end{aligned} \tag{3.18}$$

Although the total temperature and pressure at the jet are specified as constant for each case, $C_{\dot{m}}$ is not necessarily constant. $\frac{\rho_{jet}}{\rho_{\infty}}$ and $\frac{|V_{jet}|}{|V_{\infty}|}$ are functions of the local pressure and temperature at the surface of the airfoil. The average $C_{\dot{m}}$ from the 20 cases is returned to Dakota to account for this phenomenon. The $C_{\dot{m}}$ for each case is determined from the FUN3D output data.

All but the first constraint are linear functions of the design variables. The first constraint, however, relies on the output of the coupled, nonlinear, and 2nd-order Navier-Stokes equations. Therefore, the constraint is nonlinear and subject to potential discontinuities due to the presence of shock waves.

3.4 Computational Model

Like that used in the experimental study of Smith and Walker, an NACA 64A-010 airfoil with a one-degree deflection of the trailing edge is used for evaluation [9]. In reality, few, if any, operational aircraft utilize this airfoil. However, aircrafts often

start design evolution with a 6-series NACA airfoil, such as the NACA 64A-010, and modify it to meet design requirements [2].

By evaluating the NACA 64A-010 airfoil at cruise speeds near its own drag divergent Mach number, the results can be more readily extrapolated to airfoils used in practice by transonic jets. The flow conditions seen by the NACA airfoil are theoretically similar to those seen by a transonic jet flying near its own drag divergent Mach number. In both cases, a minor but inconsequential shock wave can be expected during cruise.

The airfoil and its associated grid have been generated using Pointwise grid-generation software based upon dimensions specified by Smith and Walker [9]. FUN3D is a 3-dimensional flow solver, so the 2-dimensional grid is given a pseudo 3rd dimension by extending it one cell in depth.

The airfoil is positioned such that the positive x -axis is the direction of freestream air at 0° angle of attack (right). The positive z -axis is oriented up. The y -axis is oriented along the would-be span of the wing (into the page). The top and bottom surfaces of the airfoil are specified as viscous boundaries with a no-slip condition. The outer boundary of the grid is treated as the farfield with a Riemann node boundary condition. The sides of the pseudo 3-dimensional mesh are treated as symmetry boundary conditions in the y -direction. The jet is modeled as a straight channel with inviscid walls. The end of the channel is treated as a subsonic inflow. The subsonic inflow boundary condition is defined by total temperature and total pressure ratios. Therefore, it offers a consistent representation of engine bleed over a range of flow conditions. By modeling the jet as a channel with inviscid walls, the geometry and model are simplified. The error associated with this simplification is addressed during uncertainty quantification.

The coordinates and grid dimensions are normalized by the airfoil chord. The

C-type structured mesh extends 10 chord lengths above, below, and forward of the airfoil and 20 chord lengths aft of the trailing edge.

To ensure adequate resolution of the turbulent boundary layer, its thickness (δ_{turb}) is determined using Equation 3.19 [12]

$$\delta_{turb} = \frac{0.375x}{Re_x^{1/5}} \quad (3.19)$$

In Equation 3.19, x is the distance from the leading edge and

$$Re_x = \frac{\rho_\infty V_\infty x}{\mu_\infty} = Re_{/c} x \quad (3.20)$$

Assuming fully turbulent boundary layer at the the half-chord, the boundary layer thickness (δ_{turb}) is approximately 1.0936×10^{-2} chord lengths. At least 20 vertical cells are recommended within the boundary layer to adequately resolve it [12]. Therefore, the average cell height must be less than or equal to 5.468×10^{-4} chord lengths for each mesh generated.

It is also recommended that the 1st-cell non-dimensional distance from the wall, y^+ , is less than or equal to one for each mesh. For a turbulent flat plate, y^+ is

$$y^+ = \frac{0.1707yRe_x^{0.9}}{x} \quad (3.21)$$

where y is the dimensional distance from the wall [12].

By assuming $y^+ = 1$, the maximum acceptable height of the cell nearest to the wall (Δ_1) is approximated in Equation 3.22.

$$\Delta_1 = \frac{5.86x}{Re_x^{0.9}} \quad (3.22)$$

For $Re = 2.96 \times 10^6$, Δ_1 must be less than or equal to 8.35×10^{-6} chord lengths. A

coarse mesh without an AFC jet was generated and evaluated with a Δ_1 of 8×10^{-6} chord lengths. It was then systematically refined and evaluated 4 times until a grid independent solution was reached. A grid independent solution is a solution that does not change with additional refinement. In this study, the finest mesh, with a Δ_1 of 1×10^{-6} chord lengths, serves as the baseline mesh. It is shown in Figure 3.5. Additional grid resolution is included at the trailing edge of the airfoil to account for potential separation, reattachment, and reversed flow. The baseline mesh is used to determine the baseline lift coefficient for each of the 20 cases and as the basis for geometry during optimization.

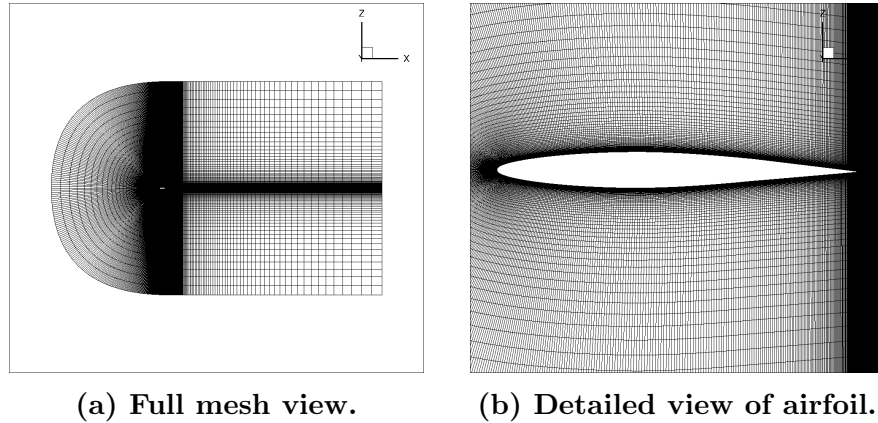


Figure 3.5. Baseline finest mesh. NACA 64A-010 airfoil, 899 X 161 nodes (143,680 cells) C-type mesh generated using Pointwise.

The location, width, and angle of the jet, specified by the Dakota optimization algorithm, are entered into a Pointwise glyph script, which autonomously modifies the baseline mesh in Figure 3.5 to include the AFC jet. The glyph script generates a jet channel with a depth five times its diameter (D_{jet}). If this default channel is too deep to fit within the airfoil, the channel is made more shallow using the Pointwise graphical user interface. Each channel has an average cell width of approximately 5×10^{-4} chord lengths. The script also provides for additional grid refinement around the jet exit to provide adequate resolution to capture its effects on the flow. An example of one of

the autonomously generated meshes with a jet is shown in Figure 3.6.

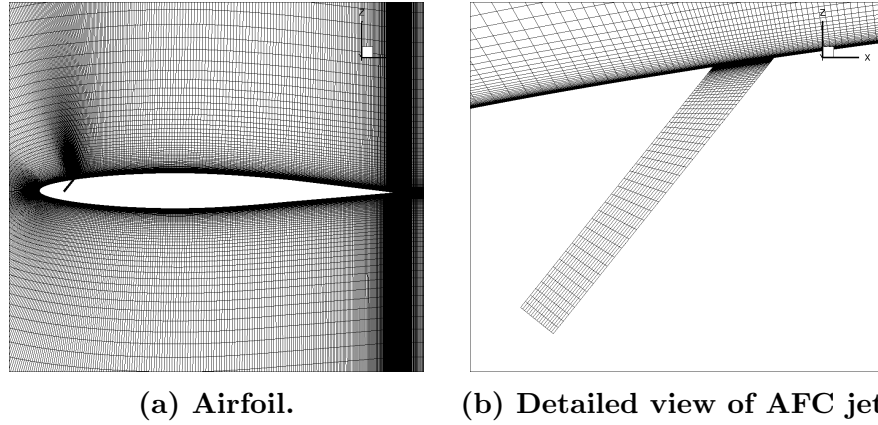


Figure 3.6. NACA 64A-010 airfoil with AFC jet (autonomously generated using the Pointwise glyph script).

Although, the Pointwise glyph script is adequate in most cases, the strategy described is not satisfactory for high-angle jets due to skewness at the mouth of the jet. When $\theta \leq 15^\circ$ or $\theta \geq 165^\circ$, an alternative strategy is utilized to design the mesh. The jet channel is separated by a connector where the jet first meets the airfoil surface. Additional connectors are added perpendicular to the airfoil at the mouth of the jet. The mesh surrounding the airfoil is modeled as three domains, one of which includes the beginning portion of the channel. The lower portion of the channel (majority) is modeled as a single domain. An example of the alternative meshing strategy is shown in Figure 3.7.

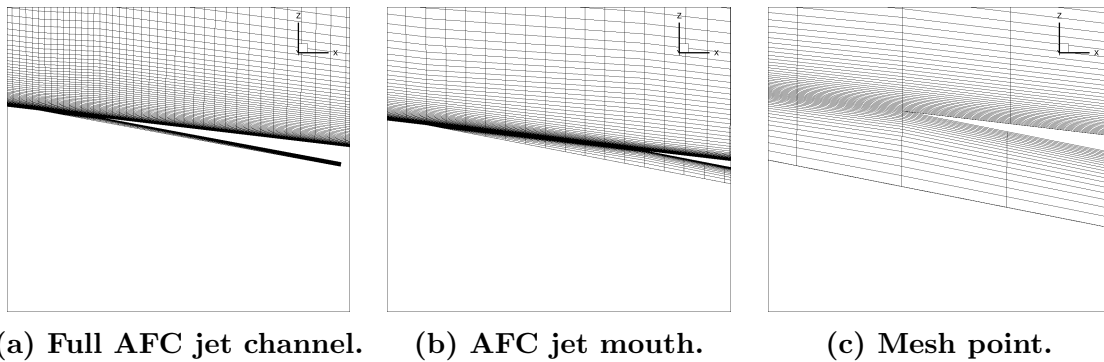


Figure 3.7. Example of alternative meshing strategy utilized when $\theta \leq 15^\circ$ or $\theta \geq 165^\circ$.

The flow is modeled about the airfoil using the 2nd-order accurate in space, steady, viscous solver in NASA FUN3D. FUN3D is an unstructured three-dimensional, implicit, node-based, finite volume, Reynolds-averaged Navier-Stokes code developed and maintained by researchers at the NASA-Langley Research Center. Steady iterations are accomplished using a “local time step pseudo time advancement scheme that is not time accurate” [16]. In this study, van Leer flux vector splitting is used for the calculation of the explicit terms. The flux limiter used is the stencil-based van Leer limiter augmented with a heuristic pressure limiter. [16]

This study uses the Spalart-Allmaras (SA) turbulence model because its baseline results most closely reflect the experimental results of Smith and Walker, when compared to Menter’s shear stress transport (SST) turbulence model and the k- ω (k- ω) turbulence model. The SA turbulence model results differ from the experimental results by 0.28-3.88%. The SST and k- ω turbulence models differ from the experimental results by 3.83-8.36% and 0.24-5.26%, respectively. The lift coefficients at 4 different angle of attack produced using each turbulence model are shown in Figure 3.8. The k- ω model also closely predicts the experimental results but appears to diverge from experimental results at high angles of attack, where the shock wave is stronger. The SA model was also utilized by Pehlivanoglu et al. in a similar computational study [8].

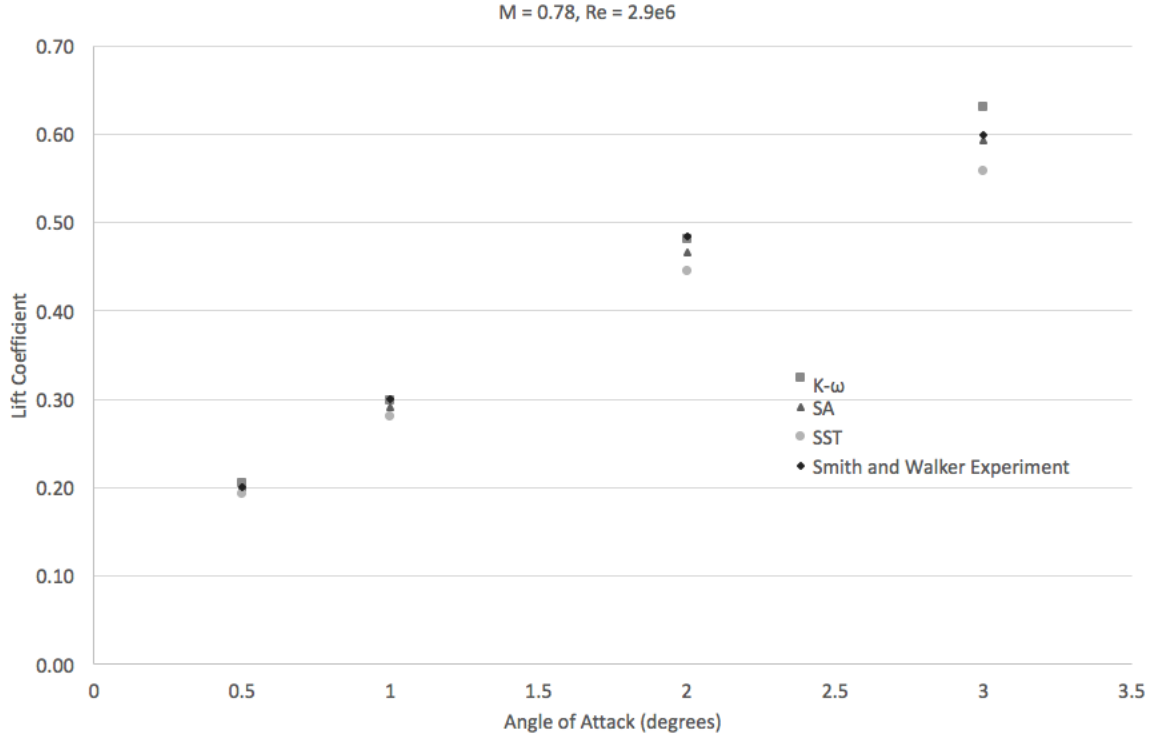


Figure 3.8. Comparison of turbulence models to experimental results of Smith and Walker [9] (baseline NACA 64A-010 airfoil).

During optimization, each evaluation is allowed to converge to a relatively constant lift coefficient. A relatively constant lift coefficient is defined, in this context, as one that does not change by more than two thousandths over 2000 or more iterations. An example of acceptable grid convergence is shown in Figure 3.9.

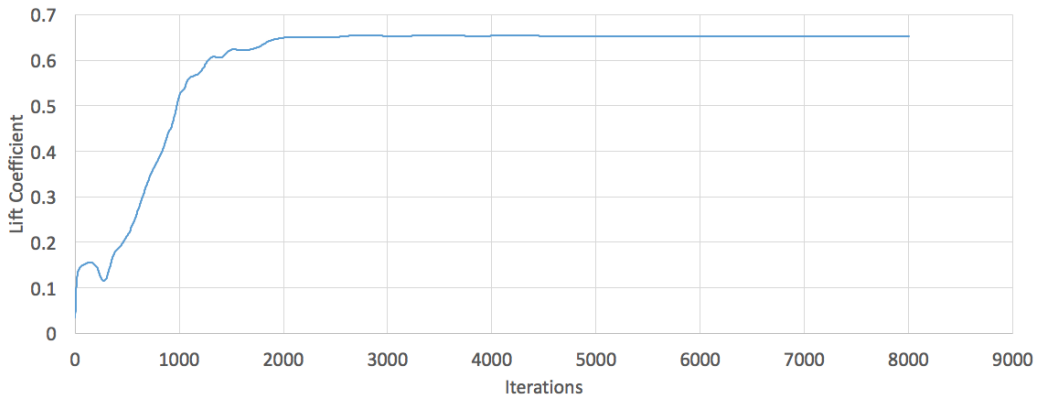


Figure 3.9. Example of acceptable grid convergence during optimization.

All computations are performed on the Department of Defense high performance computing platform, Thunder. “Thunder is an SGI ICE X System located at the AFRL DSRC. It has 3,216 standard compute nodes, 4 large-memory compute nodes, 178 Xeon Phi compute nodes, and 178 GPU compute nodes (a total of 3,576 compute nodes or 125,888 compute cores). It has 460 TBytes of memory and is rated at 5.62 peak PFLOPS.” [26] Each computation takes approximately two CPU hours on 36 cores. Approximately 250,000 core-hours were used during the course of this study.

3.5 Verification, Validation, and Uncertainty Quantification

After arriving at an optimized or significantly improved design, elements of verification, validation, and uncertainty quantification are conducted. This study does not include independent code verification. Refer to Reference [27] for verification of FUN3D with the SA turbulence model.

3.5.1 Numerical Uncertainty

Iterative error is introduced by not allowing the computations to converge completely during optimization. To quantify the numerical uncertainty due to iterative error, the improved design is allowed to fully converge as defined in Section 2.7.1. The difference between the fully converged solution and that used during optimization is used to calculate the uncertainty due to iterative error.

The mesh associated with the final design is systematically coarsened by a factor of two and systematically refined by a factor of two to produce two additional meshes. The two additional meshes are computationally evaluated to convergence using FUN3D and the lift coefficient predicted by each mesh is recorded. The uncertainty due to discretization error is estimated using the results from the three systematically refined grids, Equation 2.7, and Equation 2.8.

The uncertainty due to round off error is approximated as one percent of the discretization error. The total numerical uncertainty is approximated using Equation 2.6.

3.5.2 Model Form Uncertainty

Although Smith and Walker’s experimentation was extensive, it is unlikely that the optimized solution from this study is identical to any one experimental design tested. Therefore, it will not be possible to compare the computational results directly to experimental results. Rather, model form uncertainty is estimated by evaluating the improved design under a number of variable model parameters.

Although the SA turbulence model most closely matched the experimental results of Smith and Walker, it does not guarantee that it is providing the most accurate representation of the flow. The percent difference between each turbulence model is translated to the final design to estimate model form uncertainty. A factor of safety of 1.5 is included to account for the added momentum associated with blowing.

In this analysis, the solution is assumed to be steady. If incorrect, this assumption may introduce additional model form uncertainty. To estimate that uncertainty, the solution is reevaluated as unsteady. A number of decreasing time steps are evaluated until the solution no longer changes with decreasing time step. The difference between the steady-state solution and the time-accurate solution may be used to estimate model form uncertainty due to steady analysis.

The time-accurate solution is second-order accurate in time. Careful selection of the time step is required to adequately capture the flow characteristics. It is recommended that at least 200 time steps are evaluated over each oscillatory/time cycle [16]. In this case, the time cycle of the flow physics is unknown. It is estimated as the time required for a single particle to depart the jet and travel to the boundary

of the mesh, approximately 0.09 seconds. Therefore, the largest time step evaluated is 4.5×10^{-4} seconds. The number of iterations is adjusted for each time step to ensure that 10 complete oscillatory cycles are evaluated.

3.5.3 Input Uncertainty

The input uncertainty associated with Mach number, angle of attack, and Reynolds number are incorporated into the optimization framework. The results associated with each of the 20 cases evaluated are used to generate four unique EDF's. The span of the EDF's is representative of the input uncertainty resulting from uncertain Mach number, Reynolds number, and angle of attack.

The Mach number, Reynolds number, and angle of attack are not the only uncertain inputs, however. In reality, each of the design variables also has some degree of uncertainty. These additional sources of uncertainty are not considered in this analysis.

3.5.4 Total Predictive Uncertainty

Finally, total predictive uncertainty is estimated and graphically expressed in the form of a p-box. The center of the predictive uncertainty p-box is made up of the area between the input uncertainty EDF's. The p-box is expanded by the percent model form uncertainty and total numerical uncertainty in each direction.

IV. Results and Discussion

4.1 Optimization

Using LHS, Dakota generated 20 random designs. Of those 20 initial designs, only seven proved feasible. The remaining 13 designs did not produce a great enough change in lift to satisfy the lift constraint in Equation 3.12. The 20 initial designs and their results are included in Table 4.1. The feasible designs are highlighted in green and the infeasible designs are highlighted in red.

Table 4.1. Initial 20 designs specified by Dakota.

Design	Design Paramaters				Outputs (Returned to Dakota)	
	ℓ/s	D_{jet}/s	$T_{t,jet}/T_{\infty}$	θ (degrees)	Objective (Eq. 3.5) (* 10^{-2})	Constraint (Eq. 3.15) (* 10^{-1})
1	0.4490	0.0039	0.6788	137.1140	0.3622	0.8834
2	0.2899	0.0006	0.4807	95.9773	0.0905	0.9757
3	0.3322	0.0084	0.7289	131.4261	0.9059	0.5827
4	0.5096	0.0096	1.2472	167.1527	0.3227	1.1362
5	0.0345	0.0015	1.2983	116.4090	0.1424	0.5573
6	0.6279	0.0037	0.7850	174.7824	0.0698	1.0047
7	0.3922	0.0005	1.0789	31.3268	0.0413	0.9737
8	0.8940	0.0077	0.3732	20.6259	0.2325	0.9262
9	0.9909	0.0020	1.3941	63.3668	0.1763	-1.1388
10	0.7216	0.0056	1.4594	12.2817	0.1540	1.0043
11	0.6511	0.0111	1.3385	69.4819	0.1412	1.4880
12	0.8258	0.0091	1.1411	144.5762	0.8284	-0.0019
13	0.5875	0.0066	0.5858	157.1942	0.4982	1.3011
14	0.2475	0.0113	1.0660	7.0109	0.2364	-0.7335
15	0.9285	0.0050	0.9749	100.0965	0.6470	-1.2224
16	0.4993	0.0071	1.5533	82.2591	0.9020	1.3606
17	0.7857	0.0057	0.3616	48.9152	0.4060	1.0647
18	0.1984	0.0029	0.5460	77.8986	0.4679	-0.1941
19	0.1177	0.0103	0.8959	110.8593	0.2912	-3.2780
20	0.0657	0.0122	0.8623	38.1093	1.3864	-6.4140

Using the optimization formulation described in Chapter III, the design points quickly converged to a physically infeasible design. Within just five iterations, the prescribed D_{jet} was too small to be modeled using Pointwise. Despite returning infeasible solutions, the solver continued to prescribe designs that could not be generated.

To remedy the issue, an additional constraint was placed on D_{jet} .

$$D_{jet}/s > 1 \times 10^{-3} \quad (4.1)$$

Equation 4.1 ensures that D_{jet} is greater than 0.1% of the arc length (approximately 0.2% of the chord). In doing so, each design can be generated using Pointwise. This constraint is also physically logical. If the airfoil chord is one meter, for example, the jet diameter must exceed 2 millimeters.

Unfortunately, the algorithm continued to generate infeasible designs even with the new constraint. The designs could be generated in Pointwise, but continued to produce very little change in lift. There are a number of possible explanations for this failure. It is possible that the initial sample size was inadequate. The surrogate was generated from just 20 samples, of which only 7 were feasible. Therefore, it is likely that the feasible region is poorly represented by the surrogate. Given more time, it would be beneficial to build the initial surrogate from 100 or more Latin hypercube samples.

It is also possible that the failure is attributed to the problem formulation. The problem is designed such that the objective and constraint functions are conflicting. Small diameter jets with a low total temperature ratio generate small mass flow rate and minimize the objective. However, those jets also produce the smallest change in lift and do not meet the constraints. The optimizer appears to be placing too little emphasis on meeting the constraint.

Although time limitations do not allow for expansion of the initial sample size, the problem framework can be altered. A constraint penalty is introduced to the objective function. For each design that failed to meet the lift constraint, the objective function was increased. The updated objective function (f) including the constraint penalty is

$$f = C_m + 0.2 * \max(0, (0.1022 - \overline{\Delta C_l} + 2\sigma_{\Delta C_l})) \quad (4.2)$$

By altering the objective in this way, the separate constraint function is no longer necessary. Now, the designs that meet the lift constraint also produce the smallest objective function. The updated objective values are included in Figure 4.2. The designs that failed to meet the original constraint are highlighted in red. The smallest objective function is now returned by the feasible design with the smallest C_m .

Table 4.2. Initial 20 designs with constraint penalty.

Design	ℓ/s	D_{jet}/s	$T_{\text{t,jet}}/T_\infty$	θ (degrees)	$f (*10^{-2})$
1	0.4490	0.0039	0.6788	137.1140	2.1290
2	0.2899	0.0006	0.4807	95.9773	2.0419
3	0.3322	0.0084	0.7289	131.4261	2.0712
4	0.5096	0.0096	1.2472	167.1527	2.5951
5	0.0345	0.0015	1.2983	116.4090	1.2571
6	0.6279	0.0037	0.7850	174.7824	2.0793
7	0.3922	0.0005	1.0789	31.3268	1.9886
8	0.8940	0.0077	0.3732	20.6259	2.0849
9	0.9909	0.0020	1.3941	63.3668	0.1763
10	0.7216	0.0056	1.4594	12.2817	2.1627
11	0.6511	0.0111	1.3385	69.4819	4.1173
12	0.8258	0.0091	1.1411	144.5762	0.8284
13	0.5875	0.0066	0.5858	157.1942	3.1004
14	0.2475	0.0113	1.0660	7.0109	0.2364
15	0.9285	0.0050	0.9749	100.0965	0.6470
16	0.4993	0.0071	1.5533	82.2591	3.6232
17	0.7857	0.0057	0.3616	48.9152	2.5354
18	0.1984	0.0029	0.5460	77.8986	0.4679
19	0.1177	0.0103	0.8959	110.8593	0.2912
20	0.0657	0.0122	0.8623	38.1093	1.3864

It was also revealed during initial optimization that Dakota will prescribe designs that are non-physical despite the linear constraint expressed in Equations 3.6 and 3.7. To remedy this issue, Equations 3.6 and 3.7 are replaced by additional bounds on the location of the jet, such that ℓ is now constrained between 0.01 and 0.99 of the arclength, s . In other words, the center of the jet cannot be positioned in the final

2% of the chord. This constraint ensures the jet does not extend past the trailing edge of the airfoil and that there is enough room for the jet channel to be modeled.

The updated optimization problem is included in Equation 4.3.

$$\begin{aligned}
&\text{Minimize} && f = C_m + 0.2 * \max(0, (0.1022 - \overline{\Delta C_l} + 2\sigma_{\Delta C_l})) \\
&\text{subject to} && 0.01 \leq \ell/s \leq 0.99 \\
&&& 1 \times 10^{-3} \leq D_{jet}/s \leq 0.0125 \\
&&& 3^\circ \leq \theta \leq 177^\circ \\
&&& 0.302 \leq T_{t,jet}/T_\infty \leq 1.581
\end{aligned} \tag{4.3}$$

Using the updated optimization framework, 26 additional designs were generated and evaluated. After 26 iterations, a change to EGO was implemented to improve the optimization process. The 26 iterations before the change are shown in Figure 4.1. The designs variables are normalized by their maximum value. The objective function value generated by each design is also included. The 20 designs of the initial population from which the first surrogate was generated is not included.

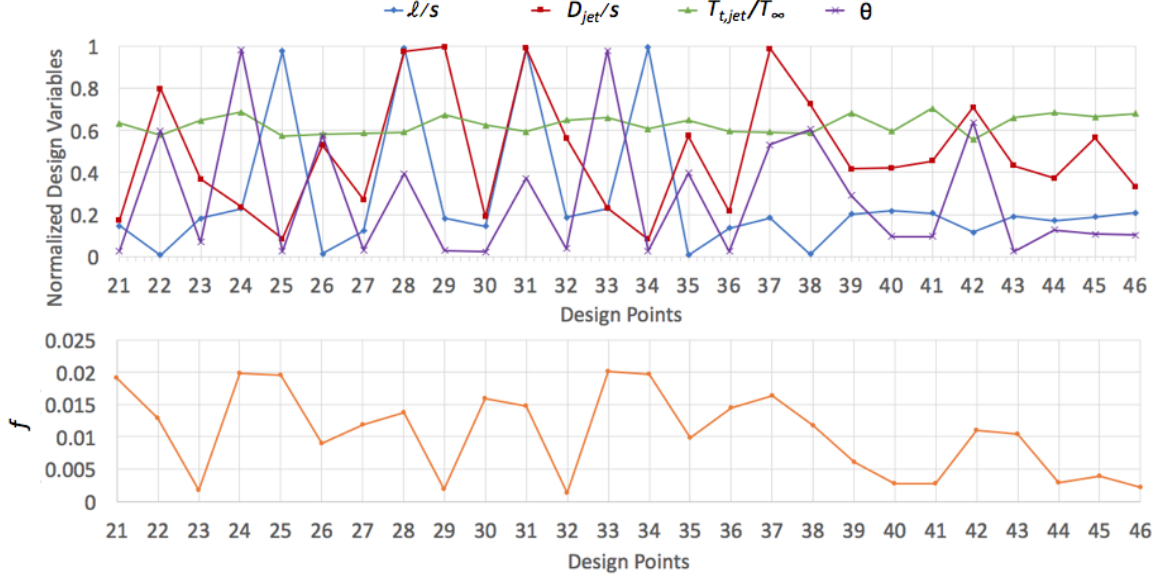


Figure 4.1. Scaled design variables (top) and resulting objective function (bottom) for first 26 design points.

Figure 4.1 shows the variability associated with the optimization. The optimal point selected from the surrogate changes drastically with each iteration. This is evidence that the surrogate is not properly modeling the solution space. If the surrogate were accurately representing the space, the optimal point selected in ensuing generations would likely remain relatively consistent or follow trends. It is clear that the initial population of 20 designs was not adequately large.

In the hopes of better capturing the solution space with the surrogate, a change to EGO is implemented. The surrogate is generated in the same way (Gaussian process) using all of the design points generated; however, the selection of the next design from that surrogate has changed.

Using EGO, 20 more designs were generated and computationally evaluated. Those designs are shown in Figure 4.2 in addition to the 26 designs generated using surrogate-based global optimization.

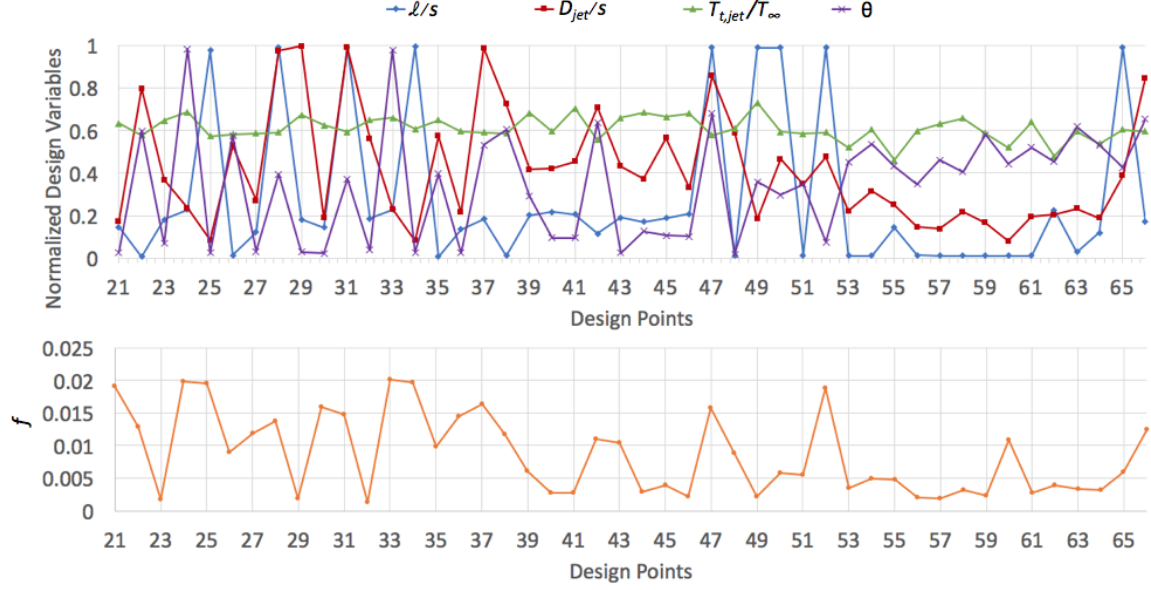


Figure 4.2. Scaled design variables and resulting objective function for 46 iterations (final 20 generated using EGO).

The final 20 EGO iterations indicate more promising trends towards an improved solution. The average objective function value has decreased from 0.0146 over the first five iterations to 0.0056 over the final five iterations. Only one design of the final 14 designs generated failed to satisfy the lift constraint. This is a significant improvement over the initial random population, where 65% of the designs failed to satisfy the constraint.

In general, the design variable values at each new design point show less variation. θ appears to be settling near 90° , or normal to the surface. Over the last 10 iterations, the standard deviation of θ is just 16.0%, compared with 57.7% from the first 10. The location also appears to be trending toward the upper surface of the trailing edge. The decreased variability between designs shows that the surrogate is beginning to better represent the solution space. However, an optimal design point has clearly not yet been determined. The design points selected by the surrogate continue to change; thus, the surrogate continues to change.

Due to time restrictions, optimization is truncated at this point. However, from

the best designs generated, some interesting results can be gained. The 20 best designs are presented in order of increasing objective value in Figure 4.3.

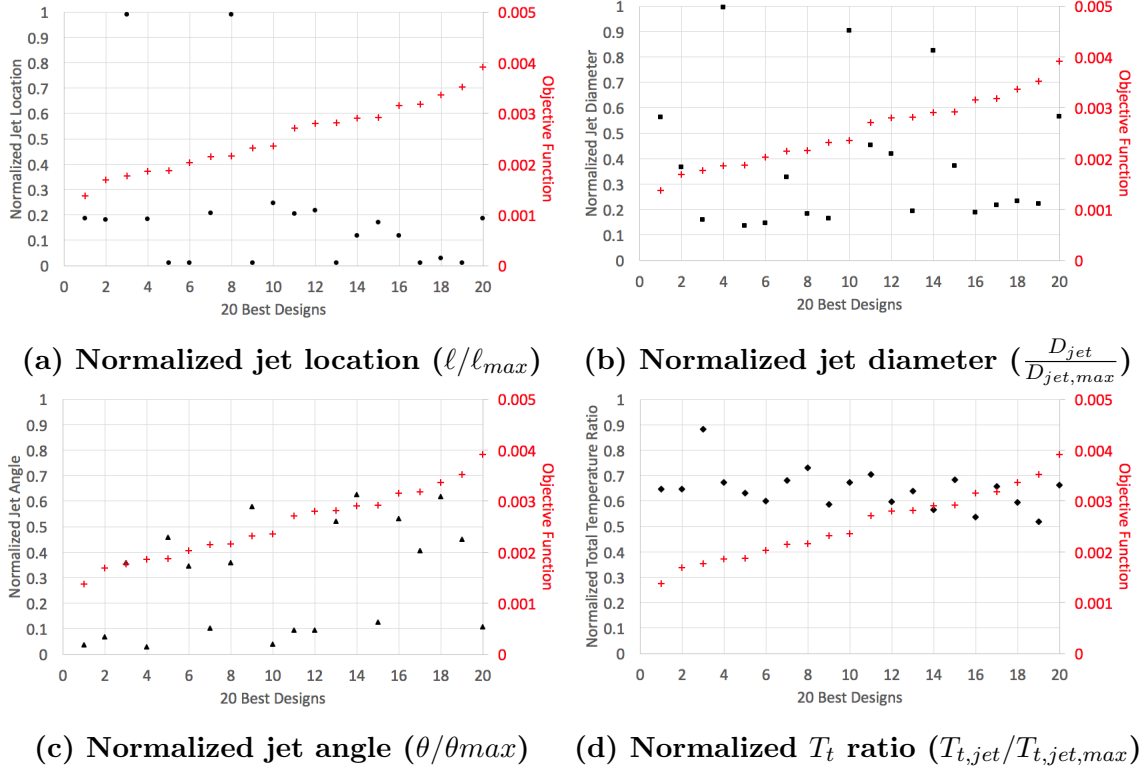


Figure 4.3. 20 best designs generated during optimization.

Although these designs are unique, they have many similarities. In 18 of the 20 designs, ℓ is less than 25% of its maximum value ($\ell/s \leq 0.25$). In 10 of the 20 designs, D_{jet} falls within 13 and 24% of its maximum value ($0.0016 \leq D_{jet}/s \leq 0.0030$). There is also a relationship between θ and D_{jet} . The 10 designs with the smallest D_{jet} in Figure 4.3 have a θ between 61° and 112° . Therefore, a majority of the best designs generated thus far have a relatively small diameter jet positioned near the trailing edge of the upper surface of the airfoil with a blowing jet oriented perpendicular to the flow. Lastly, all of the best designs generated have a total temperature ratio greater than 55% of the maximum ($T_{t,jet}/T_\infty \geq 0.8696$). These results are logical because boundary layer growth is greatest near the trailing edge. Momentum from

the jet will have the most pronounced influence here. It is likely not a coincidence that a traditional aileron is also located at the trailing edge.

4.2 Improved Design

Despite the failure of the optimization algorithm to reach an optimal solution within time restrictions, 41 of the 66 total designs produced feasible solutions. Of those 41, one design (design 9) required a C_m of just 1.763×10^{-3} . When translated to the Boeing 747 discussed in Section 3.3.3, this correlates to approximately 7.136 kg/s or just 1.14% of the mass flowing through one of its four CF6 engines.

Design 9 specifies $D_{jet}/s = 1.99 \times 10^{-3}$, $\ell/s = 0.991$, $\theta = 63.4^\circ$, and $T_{t,jet}/T_\infty = 1.394$. Although it may not be optimal, design 9 satisfies the performance constraint. If the assumptions made in this analysis are correct, then design 9 shows that a steady blowing AFC jet could effectively replace traditional control surfaces during cruise. However, further analysis with additional modeling complexity is required to confirm that the assumptions are indeed accurate.

Design 9 features a small diameter blowing jet located on the lower surface of the trailing edge of the airfoil. The design is shown in Figure 4.4 with Mach gradients and vectors.

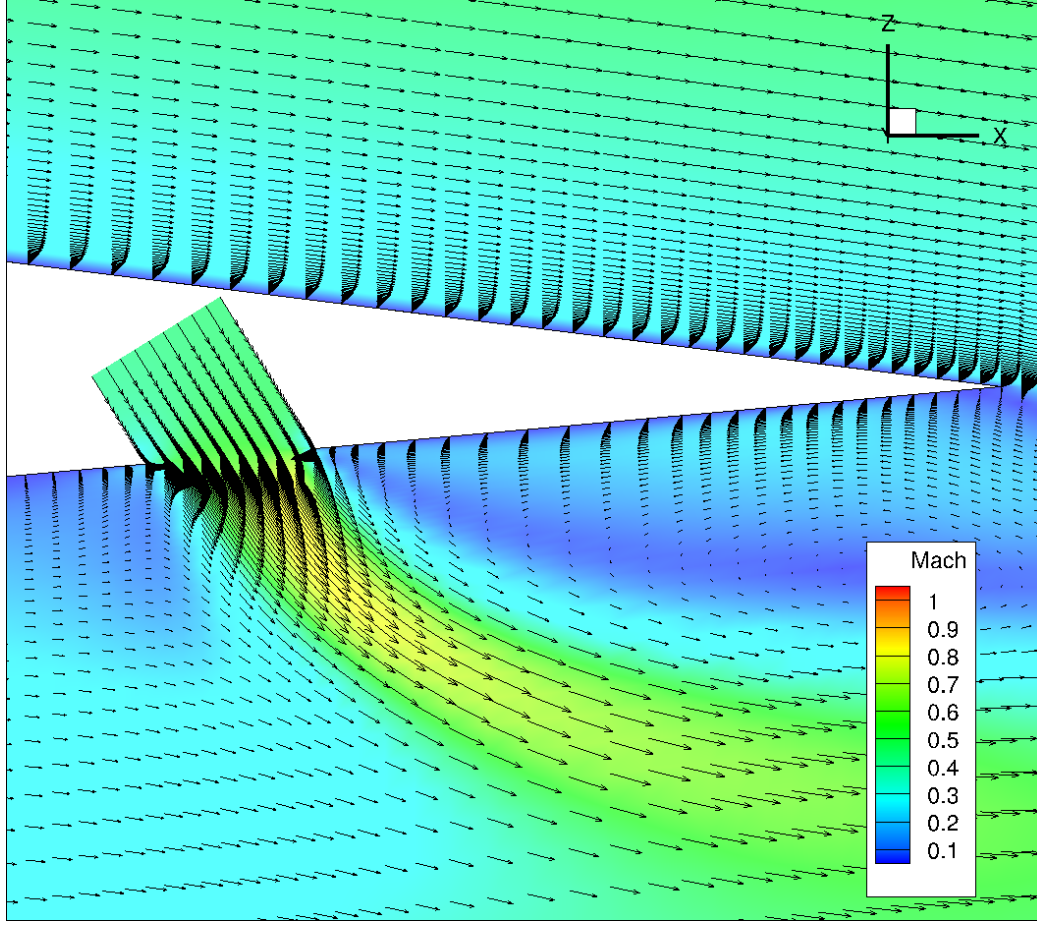


Figure 4.4. Trailing edge of design 9 ($Re = 0.95 \times 10^6$, $M = 0.75$, $\alpha = 0^\circ$, FUN3D, SA turbulence model).

The jet channel is noticeably shorter than that shown in Figure 3.6. This is a result of the limited space available near the airfoil trailing edge. The channel length was manually adjusted after execution of the Pointwise glyph script to remain geometrically feasible.

The AFC jet is positioned near the trailing edge of the airfoil. The flow about the airfoil with the AFC jet is compared to the flow about the baseline airfoil and to that of an aileron in Figure 4.5. Figure 4.5c shows an NACA 64A-010 airfoil with an aileron (final 30% of the chord) deflection of 4 degrees that has been computationally modeled using FUN3D under identical flow conditions. The freestream conditions in all three cases are the same ($Re = 0.95 \times 10^6$, $M = 0.75$, and $\alpha = 0^\circ$).

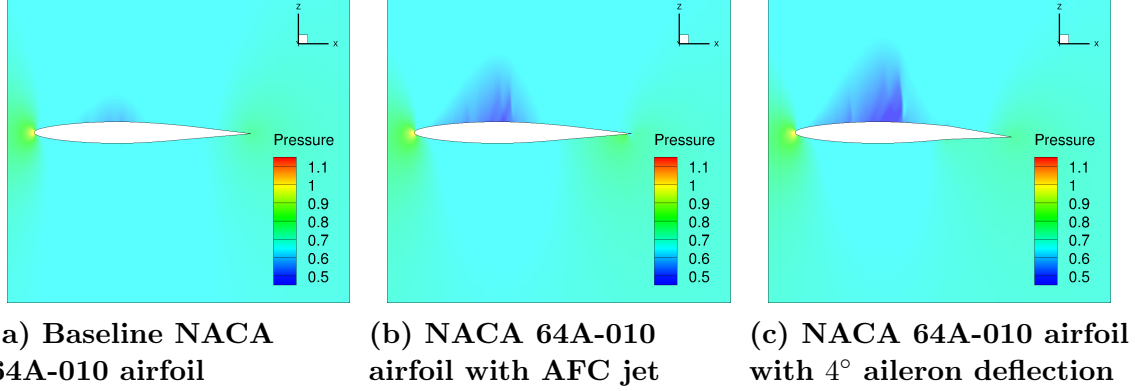


Figure 4.5. Visualization of flow about design 9 compared to the baseline airfoil and aileron ($Re = 0.95 * 10^6$, $M = 0.75$, and $\alpha = 0^\circ$).

The pressure in Figure 4.5 is in nondimensional FUN3D units. Figure 4.4 shows that the AFC jet does not act entirely like an aileron; however, Figure 4.5 shows that the overall effect is similar. The AFC jet in Figure 4.5b, like an aileron, creates an increased pressure difference between the upper and lower surface of the airfoil. With AFC jet blowing, the pressure along the upper surface of the airfoil is decreased and the pressure along the lower surface is increased. Further, for a comparable increase in lift coefficient for blowing and for a deflected aileron, the blowing case results in a somewhat weaker shock on the upper surface, which serves as another possible performance improvement. The similarity in effects of this design to a traditional aileron is convenient and not entirely surprising. A traditional aileron offers effective control over a broad range of flow conditions. Therefore, an AFC jet that can manipulate the flow in a similar manner should provide similarly robust control. Additionally, the AFC jet is more likely to provide variable control in a similar manner to typical ailerons. This idea should be analyzed further in future studies.

4.3 Uncertainty Quantification

In the absence of an optimal solution, uncertainty quantification is performed on design 9.

4.3.1 Input Uncertainty

The input uncertainty in the optimization is expressed in Figure 4.6. Four variable Mach number EDF's are shown together.

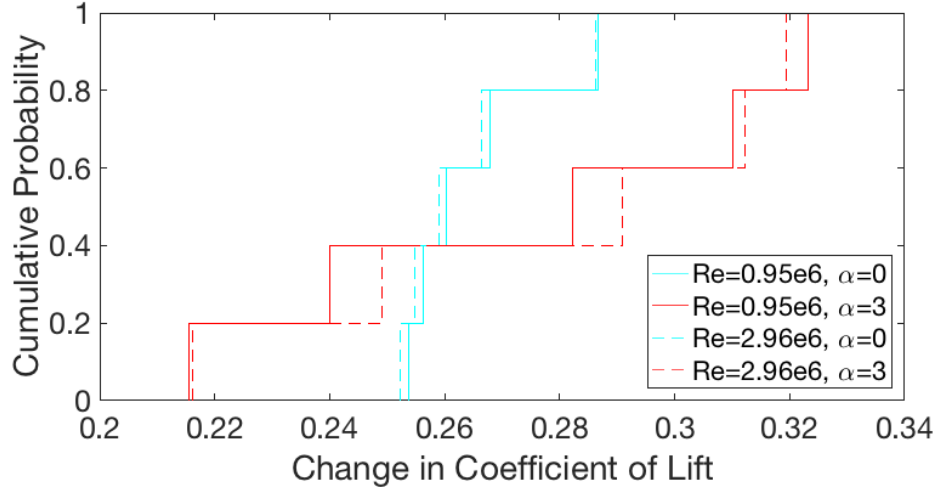


Figure 4.6. Results of input uncertainty from variable Mach number, Reynolds number, and angle of attack.

Figure 4.6 demonstrates the effects of the sources of uncertainty associated with the solution. The Reynolds number has very little impact on the effectiveness of the jet. At $\alpha = 0^\circ$, the difference in Reynolds number results in just 0.2% difference in ΔC_l . At $\alpha = 3^\circ$, the maximum difference is 3.7%. Thus, Reynolds number is not a large contributor to uncertainty.

The change in lift generated by the jet is significantly affected by variable Mach number and angle of attack. Although the mean change in lift generated at each angle of attack is similar, the distribution varies greatly. At $\alpha = 3^\circ$ and $Re = 0.95 \times 10^6$, the uncertainty due to change in Mach number from 0.712 to 0.788 is 40.0%. In comparison, the uncertainty due to change in Mach number is just 12.1% for $\alpha = 0^\circ$ and $Re = 0.95 \times 10^6$.

In each EDF, the points are plotted in order of increasing ΔC_l , not necessarily in order of increasing Mach number. As a result, the extent to which the performance

varies with Mach number is not directly captured in Figure 4.6. In this study, the performance increases with increasing Mach number for $\alpha = 0^\circ$ but decreases for $\alpha = 3^\circ$. This trend is shown in Figure 4.7 for a constant Reynolds number of 0.95×10^6 .

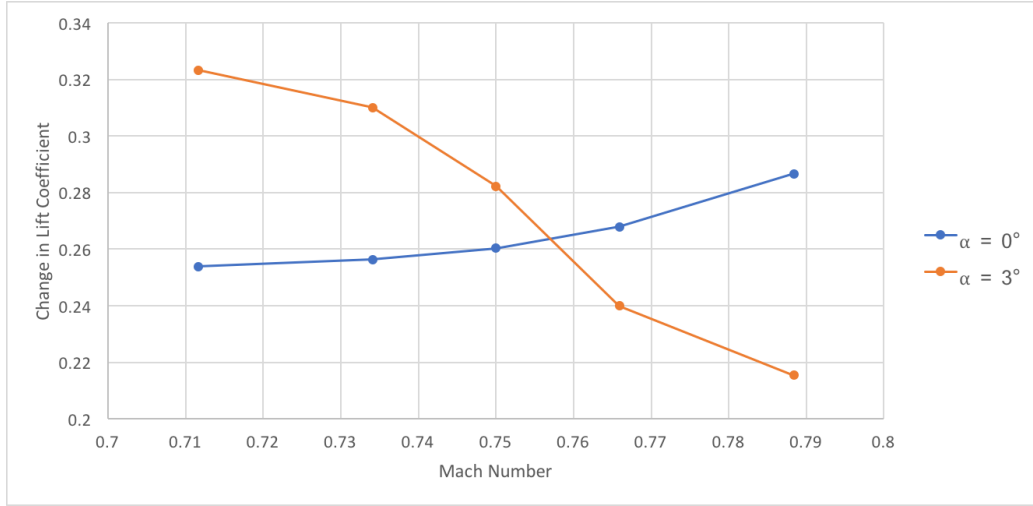


Figure 4.7. Variable performance (ΔC_l) caused by increasing Mach number for two discrete angles of attack.

There is a 24.9% decrease in performance (ΔC_l) when the angle of attack is changed from 0 to 3 degrees for Mach 0.788 freestream conditions. There also exists a Mach number between 0.75 and 0.766 where there is no difference in the change in lift generated for the two angles of attack. These interesting results can be attributed to the presence of a strong shock wave at three degrees angle of attack for $M = 0.788$ flow. Figure 4.8 shows the Mach gradient around design 9 for $\alpha = 0^\circ$ and $\alpha = 3^\circ$. The freestream Mach number and Reynolds number are the same ($M = 0.788$, $Re = 0.95 \times 10^6$).

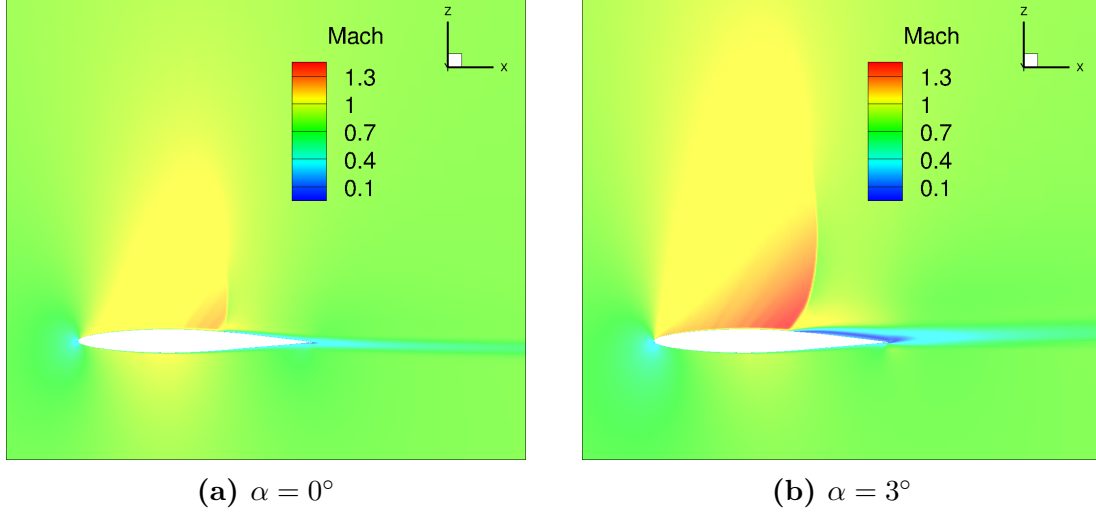


Figure 4.8. Flow visualization comparison of Design 9 for $\alpha = 0^\circ$ and $\alpha = 3^\circ$ ($M = 0.7884$, $Re = 0.95 * 10^6$, FUN3D, SA turbulence model).

The shock wave for $\alpha = 3^\circ$ is significantly stronger than that at $\alpha = 0^\circ$. There is also a greater region of separation beyond the shock when $\alpha = 3^\circ$, resulting in a much smaller lift increase due to AFC blowing.

The effectiveness of the AFC jet is heavily dependent on the angle of attack, especially at the most extreme Mach numbers. The maximum and minimum angle of attack were carefully selected to represent cruise conditions. By evaluating only the upper and lower limits during optimization, computational expenses were saved. However, the relationship between angle of attack and performance may not be linear. For each Mach and Reynolds number, $\alpha = 1.5^\circ$ is evaluated to further evaluate input uncertainty. The EDF's associated with $\alpha = 1.5^\circ$ are included in Figure 4.9.

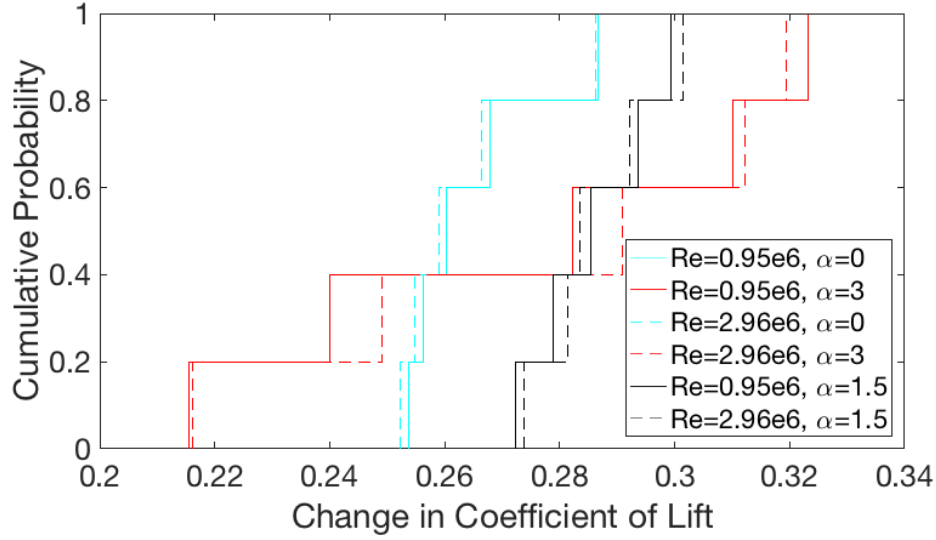


Figure 4.9. Ensemble of EDF's showing effects of input uncertainty including $\alpha = 1.5^\circ$.

The original EDF's are dashed lines in Figure 4.9 and the new ones are solid lines. The results associated with $\alpha = 1.5^\circ$ show that the relationship between angle of attack and AFC jet performance is nonlinear and potentially discontinuous. The average performance associated with $\alpha = 1.5^\circ$ is greater than that achieved when $\alpha = 0^\circ$ and $\alpha = 3^\circ$. This result is also likely attributed to the formation of a strong shock wave at $\alpha = 3^\circ$. The AFC jet generates an increase in performance as Mach number and angle of attack are increased until the shock wave reaches some undetermined strength. Although not performed in this study, a sensitivity analysis would reveal the precise Mach number at a particular angle of attack where performance begins to decrease. This relationship has important ramifications in regard to control. The performance is dependent not only on the jet, but also on the freestream conditions.

The effect of shock waves on performance demonstrate the importance of OUU. If a design were optimized strictly for Mach 0.75 at $\alpha = 0^\circ$, that design would likely not be adequate in the presence of a shock. Proper selection of uncertain input parameters is also vital. This is exemplified by analyzing design 9 at angles of attack

outside of the predicted range. Angles of attack -1° and 4° are evaluated for each Mach number. The EDF's associated with these new angles of attack are added to the input uncertainty in Figure 4.10.

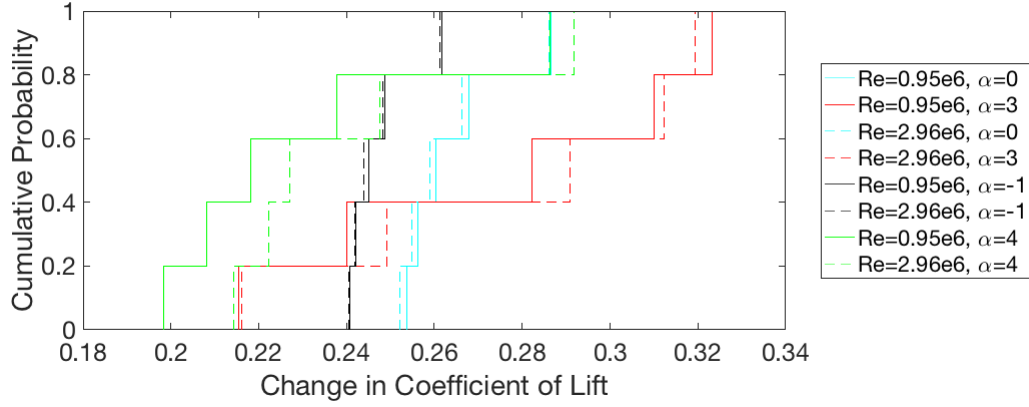


Figure 4.10. Expanded ensemble of EDFs due to input uncertainty at $\alpha = -1^\circ$ and $\alpha = 4^\circ$.

Overall performance is decreased for $\alpha = -1^\circ$ and $\alpha = 4^\circ$. The same AFC jet generates a 27.2% smaller change in lift when the angle of attack is increased from 1.5° to 4° in Mach 0.788 flow. This result demonstrates the importance of sample selection in OUU. The jet has been designed to operate in a robust manner, but only within a specified flight regime. Outside of the flight regime of interest, there is no guarantee that the design will remain useful.

4.3.2 Model Form Uncertainty

Figure 4.11 shows the results of the time accurate analyses.

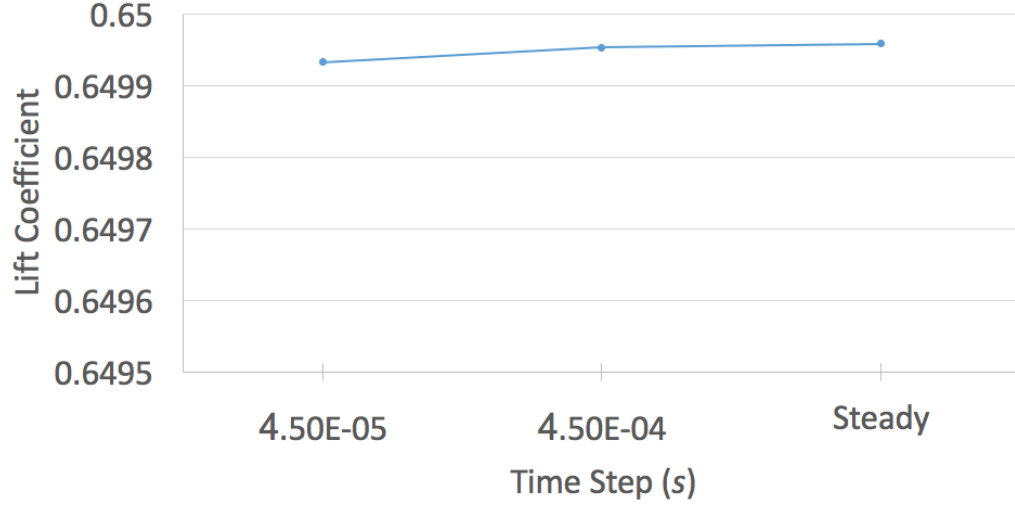


Figure 4.11. Time accurate convergence history compared to steady-state.

The y -axis scale spans only five ten-thousandths of a lift coefficient to highlight the difference between the steady state and time accurate solutions. Each of the time accurate solutions approaches a steady-state solution value. The difference between the converged steady-state solution and the smallest Δt time-accurate solution is 0.004%. Therefore, steady state is a valid assumption and does not appreciably affect the solution. The model form uncertainty due to steady-state analysis is negligible when compared to that caused by the turbulence model.

In Figure 3.8, the greatest lift coefficient is predicted by the $k-\omega$ turbulence model. The smallest is predicted by the SST turbulence model. For $\alpha = 3^\circ$, the difference between each is 12.23%. Accounting for a factor of safety of 1.5, a model form uncertainty of 18.35% is applied to the $\alpha = 3^\circ$ EDF's. For $\alpha = 0.5^\circ$, the difference is 6.45%, so 9.68% model form uncertainty is applied to the $\alpha = 0^\circ$ EDF's. Those expanded EDF's are shown in Figure 4.12. The dashed lines are the model form EDF's and correlate to the input uncertainty EDF's of the same color.

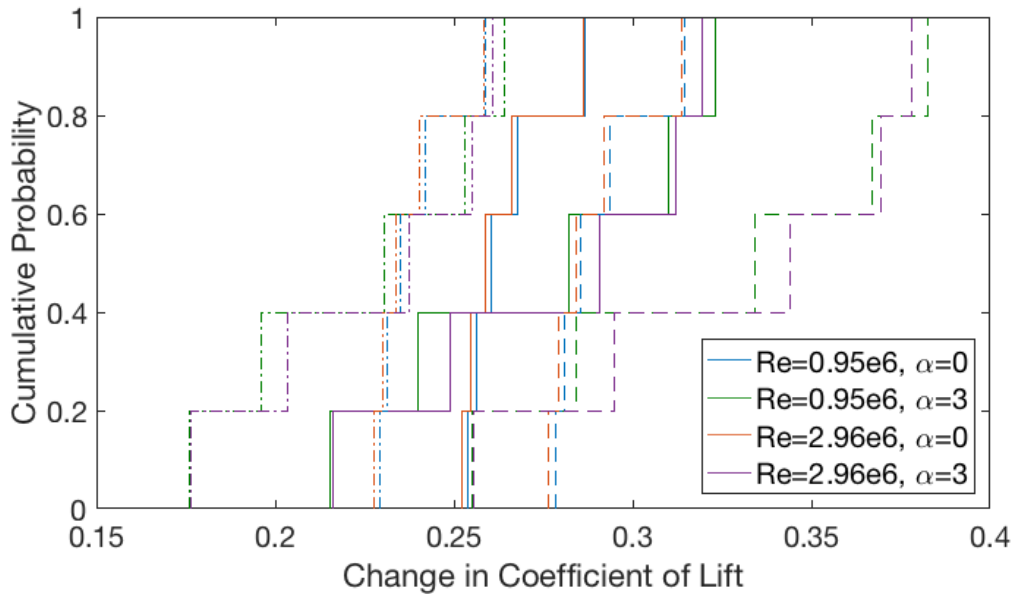


Figure 4.12. Input uncertainty EDF's expanded to include model form uncertainty.

The total model form uncertainty and input uncertainty is shown in Figure 4.13.

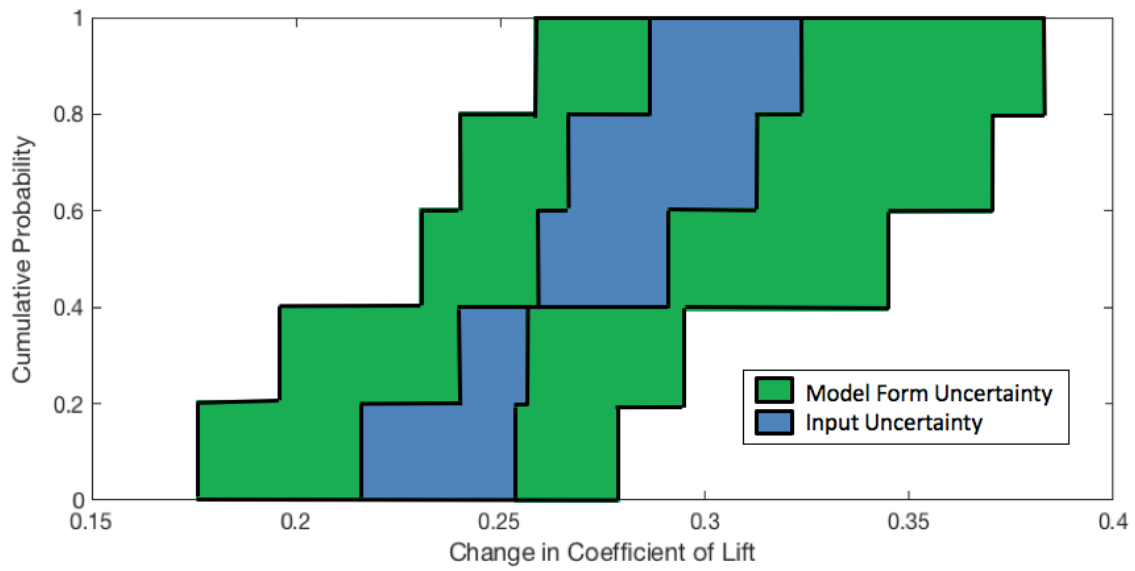


Figure 4.13. P-box including model form uncertainty and input uncertainty.

The model form uncertainty vastly outweighs the input uncertainty. The model form uncertainty is a direct result of limited experimental validation data. If a validation experiment were performed on the design of interest, model form uncertainty

could be better approximated using the area validation metric explained in Section 2.7.2. Although it is likely that the SA turbulence model does indeed best predict the flow, there is not enough data available at this time to confirm that assumption.

4.3.3 Numerical Uncertainty

Figure 4.14 shows the convergence history for design 9 under mean flow conditions ($Re = 1.955 * 10^6$, $M_\infty = 0.75$, and $\alpha = 1.5^\circ$).

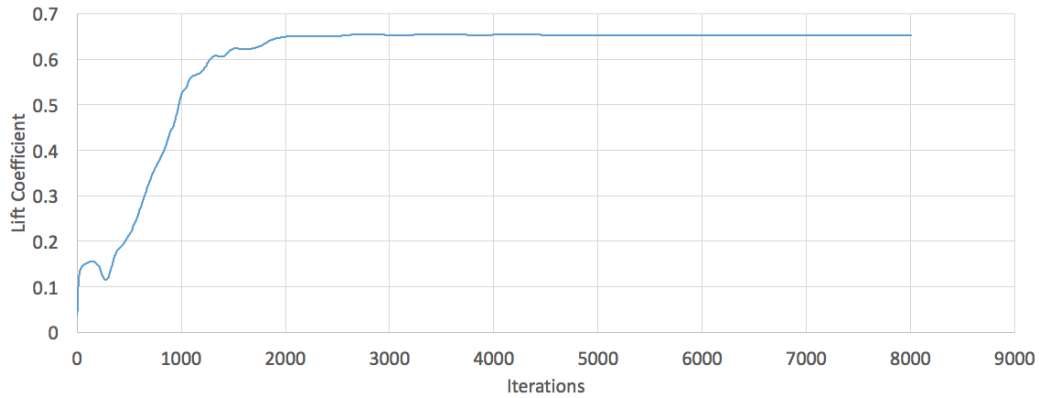


Figure 4.14. Grid convergence history (acceptable during optimization) for design 9 under mean flow conditions ($Re = 1.955 * 10^6$, $M_\infty = 0.75$, and $\alpha = 1.5^\circ$).

Although this convergence is considered acceptable during optimization, it is not fully converged and is a source of numerical error. To estimate the iterative error, the same design is allowed to fully converge over 160,000 iterations. The convergence history is included in Figure 4.15. The orange solid line shows the convergence history of the truncated solution for optimization analysis, while the blue line shows the iterative history of the fully converged solution. By greatly decreasing the y-axis scale, it is easy to see that the solution is not fully converged after just 8,000 iterations.

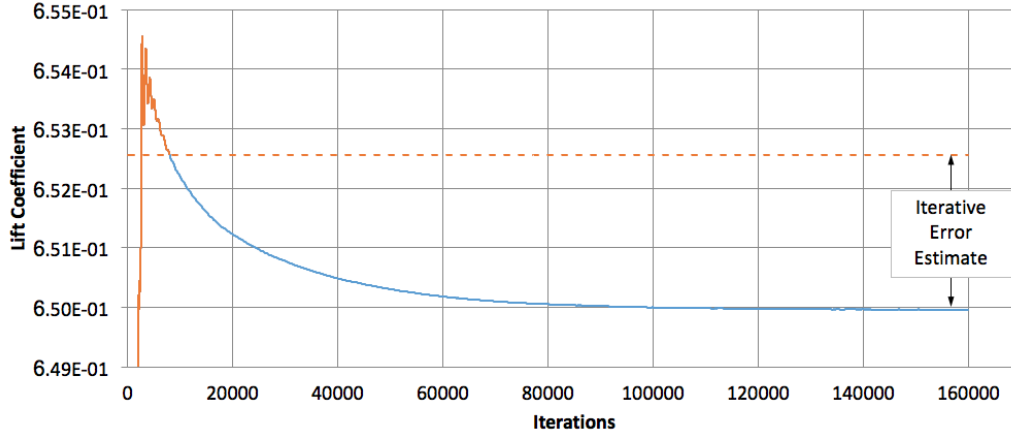


Figure 4.15. Fully converged convergence history compared to accepted convergence history for design 9 under mean flow conditions ($Re = 1.955 \times 10^6$, $M_\infty = 0.75$, and $\alpha = 1.5^\circ$).

The difference in lift coefficient between the fully converged solution in Figure 4.15 and that in Figure 3.9 is 2.61×10^{-3} ; this is shown graphically as the difference between the orange dotted line and the blue line. The uncertainty due to iterative error, U_{IT} , is 0.40%. Although the iterative error is relatively large, it is deemed acceptable during optimization. By truncating the solver before true convergence, computational expenses are cut by approximately 95%. Also, most designs generate a change in lift that is significantly greater or less than that required by the lift constraint (Equation 3.12).

The results of the systematic mesh refinement study are included in Figure 4.16. The original mesh used during optimization ($h=2$) contains 189,480 cells, the refined mesh ($h=1$) contains 757,920 cells, and the coarsened mesh ($h=4$) contains 47,370 cells.

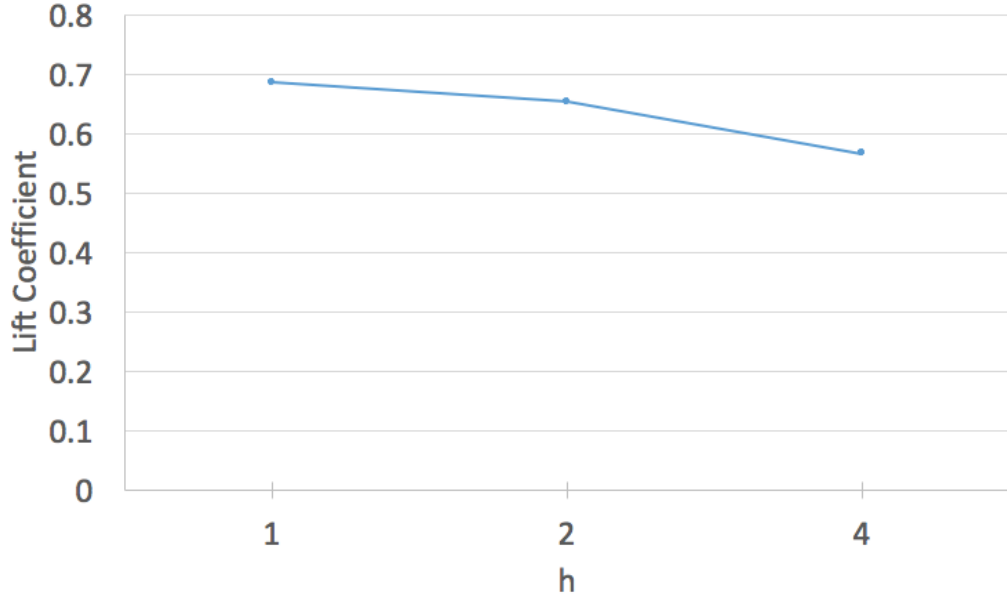


Figure 4.16. Systematic mesh refinement study under mean flow conditions ($Re = 1.955 \times 10^6$, $M_\infty = 0.75$, and $\alpha = 1.5^\circ$).

Using the results shown in Figure 4.16 and Equation 2.5, the observed order of accuracy is determined to be 1.40. There is a 30% difference between the observed order of accuracy and 2nd-order formal accuracy. This difference is due primarily to the presence of a shock wave and mesh singularities. It could also be attributed to failure of the coarsest mesh to properly capture the complex flow physics, particularly around the AFC jet. The jet is only horizontally spanned by five cells in the coarsest mesh. In Figure 4.17, the large Mach gradient estimated by those five cells is visualized.

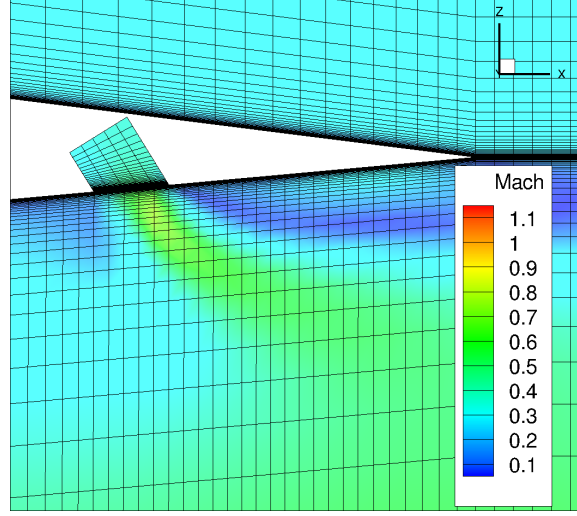


Figure 4.17. View of AFC jet (design 9) with coarsest mesh and Mach gradient.

From Equation 2.7 and the results of the systematic mesh refinement study, the discretization error is approximately 0.029. However, a factor of safety of three is used to estimate U_{DE} to account for the poor observed order of accuracy. The uncertainty due to discretization error, from Equation 2.8, is 0.087 or 13.33%. The round-off error is approximately 2.9×10^{-4} or 0.04%.

The total numerical uncertainty, from Equation 2.6, is approximately 13.77%. By adding the numerical uncertainty to the input and model form uncertainty, the total predictive uncertainty is determined. The total predictive uncertainty is shown in Figure 4.18 in the form of a p-box.

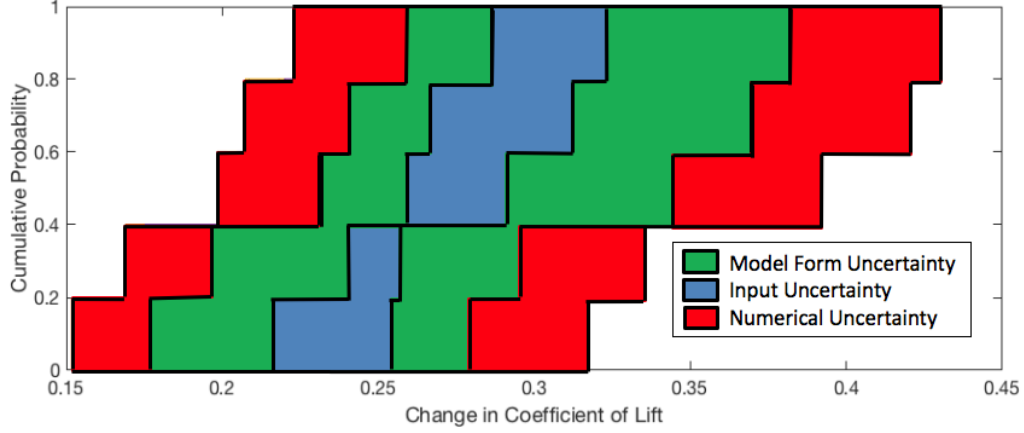


Figure 4.18. P-box representing total predictive uncertainty.

Ninety-seven percent of the numerical uncertainty can be attributed to U_{DE} . The numerical uncertainty, therefore, is primarily a result of the grid used for analysis. If the systematically refined mesh in Figure 4.16 had been used during optimization, the numerical uncertainty would be reduced to just 5.24%. However, the finer mesh required roughly eight times the number of processor core-hours to converge. At the cost of additional uncertainty, the coarser mesh was used during this analysis to expedite the optimization process and to save computational resources.

Design 9 has a predicted total uncertainty of $\pm 40.9\%$, of which $\pm 8.8\%$ is attributed to input uncertainty, $\pm 13.8\%$ to numerical uncertainty, and $\pm 18.3\%$ to model form uncertainty.

V. Conclusions and Recommendations

Although an optimal design is not determined, a number of designs may provide robust control over a range of uncertain flow conditions representing transonic cruise. Design 9 is one such design. It features a steady blowing AFC jet positioned at $\ell/s = 0.99$, with $D_{jet}/s = 1.99 \times 10^{-3}$, an angle of $\theta = 63.4^\circ$, and a total temperature ratio ($T_{t,jet}/T_{t,jet,max}$) of 1.3941. The design requires a coefficient of mass flow rate of just 1.76×10^{-3} . Translated to a Boeing 747, this is approximately 7.136 kg/s or just 1.14% of the mass flowing through one of its four CF6 engines.

The results show great promise in regards to control via a steady blowing AFC jet in transonic flight. A 2-dimensional change in lift equivalent to that required by an aileron can be generated with relatively little external mass injection. However, additional computational research should be conducted before proceeding to wind tunnel or flight test. Additional design variables and ulterior AFC mechanisms should be considered during optimization, including unsteady actuation and suction. A 3-dimensional computational study of the optimal design is also recommended. The change in drag and pitching moment affected by the AFC jet should be analyzed to ensure there are no undue consequences of its implementation. Additional uncertain inputs, such as that associated with the design variables, should also be considered. Furthermore, the determination of the mass flow rate should be automated to maintain consistency.

The OUU framework utilized here also proved effective, despite the failure to reach a converged optimum. By incorporating the performance mean and standard deviation into the optimization, many of the design points produced an adequately robust change in lift over a range of flow conditions. The framework requires a large number of function evaluations, however. It is recommended that OUU is performed with relatively inexpensive computations to narrow the design space. Once a region

of interest is determined, a more accurate computational model should be utilized to determine a robust solution with an acceptable quantity of uncertainty.

Uncertainty quantification is performed on a promising design generated early in the optimization process, design 9. The uncertainty quantification serves as an example of the process that should be completed for the optimal solution. Design 9 has a predicted total uncertainty of $\pm 40.9\%$, of which $\pm 8.8\%$ is attributed to input uncertainty, $\pm 13.8\%$ to numerical uncertainty, and $\pm 18.3\%$ to model form uncertainty. The input uncertainty captures uncertain Mach number, Reynolds number, and angle of attack during cruise. The numerical uncertainty can be accredited primarily to a relatively coarse mesh and the first-order effects of modeling a shock wave. The model form uncertainty could be decreased significantly with validation data. Without validation data, the model accuracy is difficult to estimate and a conservative uncertainty estimation must be applied. Further analysis with a refined mesh and comparison to a validation experiment would greatly reduce the total predictive uncertainty associated with the design.

Over the course of this study, multiple changes were made to successfully reach an improved design. The optimization problem was reorganized to adequately balance the objective and lift constraint into a single objective function. A minimum jet diameter and constraints on location were introduced to ensure that the designs could be physically modeled. The optimization approach was also altered from surrogate-based global optimization to efficient global optimization. Both algorithms generate surrogates using a Gaussian process, but differ in their optimization method applied to that surrogate. The ability to change the optimization framework while still utilizing the design points generated proved a valuable advantage of surrogate-based optimization. It was also shown that EGO is a good alternative to surrogate-based global optimization in cases where the initial sample is too small to generate an

adequate surrogate.

The final optimization approach is improved, though still imperfect. The initial sample from which the surrogate was generated should have been significantly larger. An initial population of 100 or more would have provided a more accurate representation of the design space. Although surrogate based optimization allows for changes in the framework quite well, the framework should remain constant for optimum performance of the algorithm. If EGO had been used from the beginning, the surrogate might better represent the solution space after the same number of iterations.

Additionally, if optimization were to be continued to a true optimal, a gradient-based approach should be considered. Surrogate-based optimization is very effective in determining a region of interest from a global design space. Once limited to a single region of interest, however, gradient-based optimization will more quickly approach the global minimum, assuming the objective function in that region is not discontinuous. This technique is known as hybrid optimization.

Appendix A. Sample FUN3D Input File

```
&project
  project_rootname = "thesis"
/

&governing_equations
  viscous_terms = "turbulent"
/

&turbulent_diffusion_models
  turbulence_model = 'sa'
/

&reference_physical_properties
  mach_number      = 0.75
  angle_of_attack  = 0
  reynolds_number  = 0.95E6
/

&code_run_control
  steps            = 30000
  stopping_tolerance = 1.0E-9
  restart_write_freq = 1000
  restart_read     = "off"
/

&raw_grid
  grid_format = "aflr3"
  data_format = "ascii"
  patch_lumping = "none"
  twod_mode = .false.
/

&global
  boundary_animation_freq = -1
  slice_freq = -1
  !volume_animation_freq = -1
/

&boundary_output_variables
  yplus = .true.
  number_of_boundaries = 4
  boundary_list = '1,3,4,5'
  mach = .true.
  vort_y = .true.
  mu_t = .true.
/

&boundary_conditions
total_pressure_ratio= 0 0 1.37295580953479441848
total_temperature_ratio= 0 0 9.139485088655185e-01
/
```

```
&inviscid_flux_method  
flux_limiter='hvanleer'  
flux_construction='vanleer'  
/  
  
&linear_solver_parameters  
meanflow_sweeps=20  
turbulence_sweeps=15  
/  
  
&nonlinear_solver_parameters  
schedule_iteration(1:2)= 1 5000  
schedule_cfl_turb(1:2)= 1 5  
schedule_cfl(1:2)= 1 5  
/
```

Appendix B. Dakota Input File: EGO

```
environment
  tabular_data
  method_pointer = 'EGO'

method
  id_method = 'EGO'
  efficient_global
  import_build_points_file = 'samples.dat'
  export_approx_points_file = 'surrogate.dat'

variables
  continuous_design = 4
  cdv_lower_bounds    0.01    0.05    0.302  0.03
  cdv_upper_bounds    0.99    0.625   1.581  1.77
  cdv_descriptor      'x'      'rad'   'Tt'   'theta'

interface,
  id_interface = 'TRUE_FN'
  system
    analysis_driver = 'script'
    parameters_file = 'params.in'
    results_file    = 'results.out'

responses,
  num_objective_functions = 1
  no_gradients
  no_hessians
```

Appendix C. Dakota Input File: Surrogate-based

```
environment
  tabular_data
    tabular_data_file = 'test.dat'
  method_pointer = 'SBGO'

method
  id_method = 'SBGO'
  surrogate_based_global
  model_pointer = 'SURROGATE'
  approx_method_pointer = 'SOGA'
  max_iterations = 10
  replace_points
  output verbose

method
  id_method = 'SOGA'
  sogas
  fitness_type merit_function
  constraint_penalty = 5
  replacement_type elitist

model
  id_model = 'SURROGATE'
  actual_model_pointer = 'TRUTH'
  surrogate global
    reuse_points all
    import_build_points_file = 'initial_20.dat'
    gaussian_process surfpack

model
  id_model = 'TRUTH'
  single
    interface_pointer = 'TRUE_FN'

variables,
  active all
  continuous_design = 4
  cdv_lower_bounds    0.0      0.0      0.302  0.03
  cdv_upper_bounds    1        0.625    1.581  1.77
  cdv_descriptor      'x'      'rad'    'Tt'    'theta'

interface,
  id_interface = 'TRUE_FN'
  system
    analysis_driver = 'post_runs_script_massflow'
    parameters_file = 'params.in'
    results_file    = 'results.out'

responses,
  objective_functions 1
  nonlinear_inequality_constraints 1
  no_gradients
  no_hessians
.
```

Appendix D. Analysis Driver BASH Script

```

sed -n 14p params.in | awk '{print $1}' > input.tmp
eval_id=$(cat input.tmp)
eval_id=$((eval_id+22) | bc -l)
echo $eval_id
rm input.tmp
while true; do
read -p "New params.in?? " yn
case $yn in
[Yy]* ) cp params.in params.$eval_id.in; break;;
[Nn]* ) break;;
* ) echo "Please answer yes or no.";;
esac
done

read -p "Set up and run design $eval_id. Then press any key. " -n1 -s

sed -n 2p params.in | awk '{print $1}' > input.tmp
x=$(cat input.tmp)
sed 's/e/'*10^/'/' input.tmp > input2.tmp          # replacing e with mathematical expression
sed 's/+/'/'/' input2.tmp > input3.tmp              # getting rid of + sign (doesn't work in bc math)
x=$(cat input3.tmp)                                  # redetermining x to be used in constraints later
rm input.tmp input2.tmp input3.tmp
x=$(echo "$x*100" | bc -l)                           # deal with the scaling

sed -n 3p params.in | awk '{print $1}' > input.tmp
radius=$(cat input.tmp)
sed 's/e/'*10^/'/' input.tmp > input2.tmp          # replacing e with mathematical expression
sed 's/+/'/'/' input2.tmp > input3.tmp              # getting rid of + sign (doesn't work in bc math)
radius=$(cat input3.tmp)                            # redetermining radius to be used to determine mom_coef later
rm input.tmp input2.tmp input3.tmp

sed -n 5p params.in | awk '{print $1}' > input.tmp
theta=$(cat input.tmp)
sed 's/e/'*10^/'/' input.tmp > input2.tmp          # replacing e with mathematical expression
sed 's/+/'/'/' input2.tmp > input3.tmp              # getting rid of + sign (doesn't work in bc math)
theta=$(cat input3.tmp)                             # redetermining radius to be used to determine mom_coef later
(needs this form for                                # math)
rm input.tmp input2.tmp input3.tmp
theta=$(echo "$theta*100" | bc -l)                  # deal with the scaling

sed -n 4p params.in | awk '{print $1}' > input.tmp
sed 's/e/'*10^/'/' input.tmp > input2.tmp          # replacing e with mathematical expression
sed 's/+/'/'/' input2.tmp > input3.tmp              # getting rid of + sign (doesn't work in bc math)
Ttjet_Tinf=$(cat input3.tmp)                       # redetermining Tt to be used in bc math
rm input.tmp input2.tmp input3.tmp                  # this is the ratio of total temperature at the jet to
                                                    # freestream temp (fun3d input)

##### determine mean change in lift #####
# for Re_2pt96
lift_change_total=0
count=0
while [ $count -lt 10 ]
do
cd /Users/welchla/Desktop/Dakota/efficient_global/$eval_id/Re_2pt96/Results/Case_$count
sed -n 121p thesis.forces | awk '{print $3}' > lift.tmp
sed 's/E/'*10^/'/' lift.tmp > lift2.tmp              # replacing e
sed 's/+/'/'/' lift2.tmp > lift3.tmp                  # getting rid of + sign
lift=$(cat lift3.tmp)
lift_baseline=$(cat /Users/welchla/Desktop/Dakota/thesis_1104/Re_2pt96/baseline/Case_$count/lift_baseline.dat)
lift_change=$(echo "$lift-$lift_baseline" | bc -l)
lift_change=${lift_change#-}                          # absolute value of change in lift
echo $lift_change > liftchange.dat                    # writing unique file for each nml
                                                    # file to be referenced later
lift_change_total=$(echo "$lift_change_total+$lift_change" | bc -l) # total change in lift
rm lift.tmp lift2.tmp lift3.tmp                       # clean up
count=$((count+1))
done

```

```

# for Re_0pt95
count=0
while [ $count -lt 10 ]
do
    cd /Users/welchla/Desktop/Dakota/efficient_global/$eval_id/Re_0pt95/Results/Case_$count
    sed -n 121p thesis.forces | awk '{print $3}' > lift.tmp
    sed 's/E/'*10^/' lift.tmp > lift2.tmp           # replacing e
    sed 's/+/'/' lift2.tmp > lift3.tmp             # getting rid of + sign
    lift=$(cat lift3.tmp)
    lift_baseline=$(cat /Users/welchla/Desktop/Dakota/thesis_1104/Re_0pt95/baseline/Case_$count/
lift_baseline.dat)
    lift_change=$(echo "$lift-$lift_baseline" | bc -l)
    lift_change=$(echo ${lift_change#-})           # absolute value of change in
lift
    echo $lift_change > liftchange.dat             # writing unique file for each
nml file to be referenced later
    lift_change_total=$(echo "$lift_change_total+$lift_change" | bc -l) # total change in lift (for
mean change in lift)
    rm lift.tmp lift2.tmp lift3.tmp               # clean up
    count=$((count+1))
done

mean_change_lift=$(echo "$lift_change_total/20" | bc -l)

##### determine std dev in lift #####
count=0
tot_diff_mean_sq=0
# for Re_0pt95
while [ $count -lt 10 ]
do
    cd /Users/welchla/Desktop/Dakota/efficient_global/$eval_id/Re_0pt95/Results/Case_$count
    lift_change=$(cat liftchange.dat)
    diff_from_mean_sq=$(echo "($mean_change_lift-$lift_change)^2" | bc -l) # calc diff btwn mean change
in lift and dchange in lift for
    # each .nml, squared
    tot_diff_mean_sq=$(echo "$tot_diff_mean_sq+$diff_from_mean_sq" | bc -l) # sum all squared for use in
samle std dev eqn
    count=$((count+1))
done

# for Re_2pt96
count=0
while [ $count -lt 10 ]
do
    cd /Users/welchla/Desktop/Dakota/efficient_global/$eval_id/Re_2pt96/Results/Case_$count
    lift_change=$(cat liftchange.dat)
    diff_from_mean_sq=$(echo "($mean_change_lift-$lift_change)^2" | bc -l) # calc diff btwn mean change
in lift and dchange in lift for
    # each .nml, square
    tot_diff_mean_sq=$(echo "$tot_diff_mean_sq+$diff_from_mean_sq" | bc -l) # sum all squared for use in
samle std dev eqn
    count=$((count+1))
done

std_dev_change_lift=$(echo "sqrt($tot_diff_mean_sq/(19))" | bc -l)

##### determine power #####
Ptjet_Pinf=$(echo "e((1.45228)+(1.4/1.4)*ln($Ttjet_Tinf/1.1125))" | bc -l)
while true; do
    read -p "Dd you already determine Mach and density for job $eval_id? " yn
    case $yn in
        [Yy]* ) cd /Users/welchla/Desktop/Dakota/efficient_global/$eval_id; Mach=$(cat Mach.dat); density=$(
cat density.dat); break;;
        [Nn]* )
            echo "What is the density at the jet inlet of test $eval_id for Case_0 Re_0pt95?"
            read density1 # density of jet over
freestream density
            echo "What is the Mach number at the jet inlet of test $eval_id for Case_0 Re_0pt95?"
            read Mach1 # Mach of jet
            echo "What is the density at the jet inlet of test $eval_id for Case_9 Re_0pt95?"
            read density2
            echo "What is the Mach number at the jet inlet of test $eval_id for Case_9 Re_0pt95?"
            read Mach2
            echo "What is the density at the jet inlet of test $eval_id for Case_0 Re_2pt96?"
            read density3
            echo "What is the Mach number at the jet inlet of test $eval_id for Case_0 Re_2pt96?"
            read Mach3
            echo "What is the density at the jet inlet of test $eval_id for Case_9 Re_2pt96?"
            read density4
            echo "What is the Mach number at the jet inlet of test $eval_id for Case_9 Re_2pt96?"
            read Mach4
    esac
done

```

```

Mach=$(echo "($Mach1+$Mach2+$Mach3+$Mach4)/4" | bc -l)
density=$(echo "($density1+$density2+$density3+$density4)/4" | bc -l)
echo $density > /Users/welchla/Desktop/Dakota/efficient_global/$eval_id/density.dat # save the density
echo $Mach > /Users/welchla/Desktop/Dakota/efficient_global/$eval_id/Mach.dat # save the Mach number
break;;
* ) echo "Please answer yes or no.;;"
esac
done

diameter=$(echo "$radius/100*4.16" | bc -l) # diameter of jet in grid units
diameter_end=$(echo "$diameter*s($theta*3.14159/180)" | bc -l) # diameter of the end of the jet
massflow=$(echo "$density*$diameter_end*$Mach" | bc -l) # calculating mass flow
massflow=$(echo "$massflow*100" | bc -l) # scaling massflow

##### write results file #####
constraint1=$(echo "0.102-$mean_change_lift+(2*$std_dev_change_lift)" | bc -l) # constraint in std form
constraint1=$(echo "$constraint1*10" | bc -l) # scaling constraint

if (( $(bc <<< "$constraint1 > 0" | bc) ))
then
massflow=$(echo "$massflow+(2*$constraint1)" | bc -l)
fi

while true; do
read -p "Write new results.out file for $eval_id ? " yn
case $yn in
[Yy]* )
echo 'mean' $mean_change_lift 'stdev' $std_dev_change_lift > /Users/welchla/Desktop/Dakota/efficient_global/
mean_stdev.dat.$eval_id
echo 'constraint' $constraint1 > /Users/welchla/Desktop/Dakota/efficient_global/constraint.$eval_id.dat
echo $massflow > /Users/welchla/Desktop/Dakota/efficient_global/results.$eval_id.out
# writing and saving a local copy of results.out

break;;
[Nn]* ) break;;
* ) echo "Please answer yes or no.;;"
esac
done

cp /Users/welchla/Desktop/Dakota/efficient_global/results.$eval_id.out /Users/welchla/Desktop/Dakota/efficient_global/
results.out # generating results.out file for dakota

read -p "Check if it worked then press any key. " -n1 -s

```

Appendix E. Sample Pointwise Glyph Script

```
Pointwise V17.3R5 Journal file - Tue Nov  8 08:24:53 2016

package require PWI_Glyph 2.17.3

pw::Application setUndoMaximumLevels 5
pw::Application reset
pw::Application markUndoLevel {Journal Reset}

pw::Application clearModified

pw::Application reset -keep Clipboard
set _TMP(mode_5) [pw::Application begin ProjectLoader]
  $_TMP(mode_5) initialize {/Users/welchla/Documents/Pointwise/1102_jet_check.pw}
  $_TMP(mode_5) setAppendMode false
  $_TMP(mode_5) load
$_TMP(mode_5) end
unset _TMP(mode_5)
pw::Application resetUndoLevels
pw::Application markUndoLevel {Open}

pw::Display resetView +X
set _TMP(mode_6) [pw::Application begin Create]
  set _DM(1) [pw::GridEntity getByName "dom-6"]
  set _DM(2) [pw::GridEntity getByName "dom-3"]
  set _DM(3) [pw::GridEntity getByName "dom-5"]
  set _DM(4) [pw::GridEntity getByName "dom-1"]
  set _TMP(PW_2) [pw::FaceStructured createFromDomains [list $_DM(1) $_DM(2) $_DM(3) $_DM(4)]]
  set _TMP(face_4) [lindex $_TMP(PW_2) 0]
  set _TMP(face_5) [lindex $_TMP(PW_2) 1]
  set _TMP(face_6) [lindex $_TMP(PW_2) 2]
  unset _TMP(PW_2)
  set _TMP(extStrBlock_1) [pw::BlockStructured create]
  $_TMP(extStrBlock_1) addFace $_TMP(face_4)
  set _TMP(extStrBlock_2) [pw::BlockStructured create]
  $_TMP(extStrBlock_2) addFace $_TMP(face_5)
  set _TMP(extStrBlock_3) [pw::BlockStructured create]
  $_TMP(extStrBlock_3) addFace $_TMP(face_6)
$_TMP(mode_6) end
unset _TMP(mode_6)
set _TMP(mode_7) [pw::Application begin ExtrusionSolver [list $_TMP(extStrBlock_1) $_TMP(extStrBlock_2)
$_TMP(extStrBlock_3)]]
  $_TMP(mode_7) setKeepFailingStep true
  $_TMP(extStrBlock_1) setExtrusionSolverAttribute Mode Translate
  $_TMP(extStrBlock_2) setExtrusionSolverAttribute Mode Translate
  $_TMP(extStrBlock_3) setExtrusionSolverAttribute Mode Translate
  $_TMP(extStrBlock_1) setExtrusionSolverAttribute TranslatedDirection {1 0 0}
  $_TMP(extStrBlock_2) setExtrusionSolverAttribute TranslatedDirection {1 0 0}
  $_TMP(extStrBlock_3) setExtrusionSolverAttribute TranslatedDirection {1 0 0}
  set _BL(1) [pw::GridEntity getByName "blk-1"]
  set _BL(2) [pw::GridEntity getByName "blk-2"]
  set _BL(3) [pw::GridEntity getByName "blk-3"]
  $_TMP(extStrBlock_1) setExtrusionSolverAttribute TranslatedDirection {0 1 0}
  $_TMP(extStrBlock_2) setExtrusionSolverAttribute TranslatedDirection {0 1 0}
  $_TMP(extStrBlock_3) setExtrusionSolverAttribute TranslatedDirection {0 1 0}
  $_TMP(extStrBlock_1) setExtrusionSolverAttribute TranslatedDistance 1
  $_TMP(extStrBlock_2) setExtrusionSolverAttribute TranslatedDistance 1
  $_TMP(extStrBlock_3) setExtrusionSolverAttribute TranslatedDistance 1
  $_TMP(mode_7) run 1
$_TMP(mode_7) end
unset _TMP(mode_7)
unset _TMP(extStrBlock_3)
unset _TMP(extStrBlock_2)
unset _TMP(extStrBlock_1)
pw::Application markUndoLevel {Extrude, Translate}

unset _TMP(face_6)
unset _TMP(face_5)
unset _TMP(face_4)
pw::Display resetView +Y
set _DM(5) [pw::GridEntity getByName "dom-2"]
set _DM(6) [pw::GridEntity getByName "dom-4"]
set _DM(7) [pw::GridEntity getByName "dom-13"]
set _DM(8) [pw::GridEntity getByName "dom-7"]
set _DM(9) [pw::GridEntity getByName "dom-8"]
set _DM(10) [pw::GridEntity getByName "dom-9"]
set _DM(11) [pw::GridEntity getByName "dom-10"]
```

```

set _DM(12) [pw::GridEntity getByName "dom-11"]
set _DM(13) [pw::GridEntity getByName "dom-12"]
set _DM(14) [pw::GridEntity getByName "dom-14"]
set _DM(15) [pw::GridEntity getByName "dom-16"]
set _DM(16) [pw::GridEntity getByName "dom-18"]
set _DM(17) [pw::GridEntity getByName "dom-20"]
set _DM(18) [pw::GridEntity getByName "dom-21"]
set _DM(19) [pw::GridEntity getByName "dom-22"]
set _DM(20) [pw::GridEntity getByName "dom-23"]
set _TMP(PW_3) [pw::BoundaryCondition getByName "Unspecified"]
set _TMP(PW_4) [pw::BoundaryCondition getByName "left_side"]
set _TMP(PW_5) [pw::BoundaryCondition getByName "right_side"]
set _TMP(PW_6) [pw::BoundaryCondition getByName "farfield"]
set _TMP(PW_7) [pw::BoundaryCondition getByName "airfoil"]
set _TMP(PW_8) [pw::BoundaryCondition getByName "jet"]
$_TMP(PW_4) apply [list [list $_BL(2) $_DM(4)]]
pw::Application markUndoLevel {Set BC}

$_TMP(PW_8) apply [list [list $_BL(2) $_DM(18)]]
pw::Application markUndoLevel {Set BC}

$_TMP(PW_8) setPhysicalType -usage CAE {Inflow (fixed)}
pw::Application markUndoLevel {Change BC Type}

$_TMP(PW_7) apply [list [list $_BL(3) $_DM(14)] [list $_BL(3) $_DM(13)]]
pw::Application markUndoLevel {Set BC}

set _TMP(PW_9) [pw::BoundaryCondition create]
pw::Application markUndoLevel {Create BC}

set _TMP(PW_10) [pw::BoundaryCondition getByName "bc-7"]
unset _TMP(PW_9)
$_TMP(PW_10) setPhysicalType -usage CAE {Wall (inviscid)}
pw::Application markUndoLevel {Change BC Type}

$_TMP(PW_10) setName "jet_wall"
pw::Application markUndoLevel {Name BC}

$_TMP(PW_10) apply [list [list $_BL(2) $_DM(17)] [list $_BL(2) $_DM(19)]]
pw::Application markUndoLevel {Set BC}

$_TMP(PW_5) apply [list [list $_BL(3) $_DM(16)] [list $_BL(2) $_DM(20)] [list $_BL(1) $_DM(12)]]
pw::Application markUndoLevel {Set BC}

$_TMP(PW_6) apply [list [list $_BL(3) $_DM(15)] [list $_BL(1) $_DM(8)] [list $_BL(1) $_DM(9)] [list $_BL(1)
$_DM(11)] [list $_BL(1) $_DM(10)]]
pw::Application markUndoLevel {Set BC}

unset _TMP(PW_3)
unset _TMP(PW_4)
unset _TMP(PW_5)
unset _TMP(PW_6)
unset _TMP(PW_7)
unset _TMP(PW_8)
unset _TMP(PW_10)
pw::Display resetView +Y
pw::Application save {/Users/welchla/Documents/Pointwise/1102_jet_generated.pw}
set _TMP(mode_8) [pw::Application begin CaeExport [pw::Entity sort [list $_BL(1) $_BL(2) $_BL(3) $_DM(3)
$_DM(1) $_DM(2) $_DM(4) $_DM(5) $_DM(6) $_DM(8) $_DM(9) $_DM(10) $_DM(11) $_DM(12) $_DM(13) $_DM(7)
$_DM(14) $_DM(15) $_DM(16) $_DM(17) $_DM(18) $_DM(19) $_DM(20)]]]
$_TMP(mode_8) initialize -type CAE {/Users/welchla/Desktop/Dakota/thesis_1104/thesis}
$_TMP(mode_8) setAttribute FileFormat ASCII
$_TMP(mode_8) setAttribute FilePrecision Double
if {![$_TMP(mode_8) verify]} {
    error "Data verification failed."
}
$_TMP(mode_8) write
$_TMP(mode_8) end
unset _TMP(mode_8)

```

Appendix F. Sample PACMAN Input File

```
&settings_global
  wait_time = 600
  wait_silent = .false.
  proj_overwrite = .false.
  ! alternative_proj_name = 'Test1'
  ! alternative_case_name = 'Run'
  convert_ascii_ugrid = .false.

  run_local = .false.

  pacman_sleep = 0
/

&adoc_global
  run_single_case = .false.

  setup_only = 'submit'

  matrix_file = ''

  design_type = 'fullfact' !'2levelfull'

  ! !!!!! Currently supported design variables !!!!!!!
  var_aoa = 0 3
  ! var_yaw = 10 11
  var_ma = .7116 .7341 .75 .7659 .7884
  ! var_temperature = 400 500 600
  ! var_viscous = "inviscid" "laminar" "turbulent"
  !
  ! var_turb_model = 'sa' 'sst'
  ! var_massflow_set_out = 77 78 79
  ! var_proj_name = 'Grid1' 'Grid2' 'Grid3'
  ! var_ptot_bc = 1.0 1.05
  ! var_ttot_bc = 1.0 1.01
  ! var_static_prat_bc = 1.0 1.4
  ! var_time_step = 0.01 0.2
  ! var_steps = 1000 2000
/

&server_info
  file_overwrite = .true.

  number_of_servers = 1

  queue = 'standard'

  max_num_nodes = 4
  min_num_nodes = 1
  bad_nodes_fudge = 0

  max_num_hours = 2

  username(1) = "welchla"
  server_name(1) = 'thunder'
  home_dir(1) = '/home/welchla/data/Re_0pt95/ego1'
  !alt_fun3d_dir(1) = 'special/fun3d/install'
/

&fun3d
  adapt = .false.
  adapt_type = 'feature' !'adjoint'
  adapt_iteration = 5

  sensitivity = .false.
  sensitivity_type = 'complex' !'adjoint'

  use_local_restart = .false.

  use_local_case_restart = .false.

  get_tecplot_volume = .true.
  get_tecplot_boundary = .true.
  get_tecplot_sampling = .false.
  get_tecplot_slice = .true.
```

```

get_info_out = .true.
get_cgns = .true.
get_tecplot_all = .true.
get_fun3d_bad_hex = .true.
get_fun3d_nan = .true.

get_resid_and_force = .true.

get_fieldview_file = .true.

get_fun3d_flow = .true.

get_fun3d_moving_body = .false.

get_case_inputs = .true.

get_fun3d_sensitivity = .false.

!clo(1) = '--animation_freq -1'
!clo(2) = '--moving_grid'
!clo(3) = '--aeroelastic_internal'
!clo(4) = '--temporal_err_control 1.0'
/

```

Appendix G. BASH Script for Generating New Grid

```

echo "Which grid would you like to generate?"
read par_num
##### make changes to pointwise script from Dakota output variables #####

sed -n 2p params.$par_num.in | awk '{print $1}' > input.tmp
x=$(cat input.tmp)
sed 's/e/*10^/' input.tmp > input2.tmp          # replacing e
sed 's/+/'/' input2.tmp > input3.tmp            # getting rid of + sign
x=$(cat input3.tmp)                             # redetermining x
x=$(echo "$x*100" | bc -l)
sed 's/replacePercent/'$x'/' /Users/welchla/Documents/Pointwise/makejet_1104_change.glf > /Users/
welchla/Documents/Pointwise/makejet_1104_new.glf
rm input.tmp input2.tmp input3.tmp

sed -n 3p params.$par_num.in | awk '{print $1}' > input.tmp
radius=$(cat input.tmp)
sed 's/replaceRadius/'$radius'/' /Users/welchla/Documents/Pointwise/makejet_1104_new.glf > /Users/
welchla/Documents/Pointwise/makejet_1104_new2.glf
sed 's/e/*10^/' input.tmp > input2.tmp          # replacing e
sed 's/+/'/' input2.tmp > input3.tmp            # getting rid of + sign
radius=$(cat input3.tmp)                       # redetermining radius
rm input.tmp input2.tmp input3.tmp

sed -n 5p params.$par_num.in | awk '{print $1}' > input.tmp
theta=$(cat input.tmp)
sed 's/e/*10^/' input.tmp > input2.tmp          # replacing e
sed 's/+/'/' input2.tmp > input3.tmp            # getting rid of + sign
theta=$(cat input3.tmp)                       # redetermining radius
theta=$(echo "$theta*100" | bc -l)
sed 's/replaceTheta/'$theta'/' /Users/welchla/Documents/Pointwise/makejet_1104_new2.glf > /Users/
welchla/Documents/Pointwise/makejet_1104_new3.glf
rm input.tmp input2.tmp input3.tmp

scale=$(echo "s($theta*3.14159/180)/5" | bc -l)
sed 's/scaleFactor/'$scale'/' /Users/welchla/Documents/Pointwise/makejet_1104_new3.glf > /Users/welchla/
Documents/Pointwise/makejet_1104_new4.glf
sed 's/scaleFactor/'$scale'/' /Users/welchla/Documents/Pointwise/makejet_1104_new4.glf > /Users/welchla/
Documents/Pointwise/makejet_1104_new5.glf
sed 's/scaleFactor/'$scale'/' /Users/welchla/Documents/Pointwise/makejet_1104_new5.glf > /Users/welchla/
Documents/Pointwise/makejet_1104.glf

##### generate new grid #####
echo 'Generating 2D grid...'
/Applications/Pointwise/PointwiseV17.3R5/macosx/Pointwise.app/Contents/MacOS/tclsh8.5 /Users/welchla/
Documents/Pointwise/makejet_1104.glf
read -p "2D grid is generated (1102_jet_check.pw). Check it out, update and save it, then press any
key. " -n1 -s
echo 'Generating 3D grid...'
/Applications/Pointwise/PointwiseV17.3R5/macosx/Pointwise.app/Contents/MacOS/tclsh8.5 /Users/welchla/
Documents/Pointwise/makejet_1104_3d_many.glf

#### copy grid files to proper folders and run all of the .nml files
# copy grid files to Re_2pt96
cp /Users/welchla/Desktop/Dakota/generate_all_grids/thesis.mapbc /Users/welchla/Desktop/Dakota/
efficient_global/$par_num/Re_2pt96/Description/thesis.mapbc
cp /Users/welchla/Desktop/Dakota/generate_all_grids/thesis.ugrid /Users/welchla/Desktop/Dakota/
efficient_global/$par_num/Re_2pt96/Description/thesis.ugrid
# copy grid files to Re_0pt95
cp /Users/welchla/Desktop/Dakota/generate_all_grids/thesis.mapbc /Users/welchla/Desktop/Dakota/
efficient_global/$par_num/Re_0pt95/Description/thesis.mapbc
cp /Users/welchla/Desktop/Dakota/generate_all_grids/thesis.ugrid /Users/welchla/Desktop/Dakota/
efficient_global/$par_num/Re_0pt95/Description/thesis.ugrid

echo 'CHANGE the mapbc file to have BC 7011'
echo 'CHANGE the pacman.nml file to have coliny$par_num'

```

Appendix H. BASH Script for Changing FUN3D Input File

```
echo "Which fun3d.nml would you like to change?"
read par_num
sed -n 4p params.$par_num.in | awk '{print $1}' > input.tmp
Tt=$(cat input.tmp) # this is the ratio of total temperature at the jet to freestream temp (fun3d input)

# change Re_0pt95 .nml file
sed 's/Tt_ratio/'$Tt'/' /Users/welchla/Desktop/Dakota/soga_con/fun3d.change > /Users/welchla/
Desktop/Dakota/efficient_global/$par_num/Re_0pt95/Description/fun3d.temp

# change Re_2pt96 .nml file
sed 's/Tt_ratio/'$Tt'/' /Users/welchla/Desktop/Dakota/soga_con/fun3d296.change > /Users/
welchla/Desktop/Dakota/efficient_global/$par_num/Re_2pt96/Description/fun3d.temp

sed 's/e/'*10^'/' input.tmp > input2.tmp          # replacing e
sed 's/+/'/'/' input2.tmp > input3.tmp            # getting rid of + sign
Tt=$(cat input3.tmp)                               # redetermining Tt
rm input.tmp input2.tmp input3.tmp

Pt=$(echo "e(l(1.45228)+(.4/1.4)*l($Tt/1.1125))" | bc -l) # determining Pt
echo $Pt > input.tmp
echo $Pt

# change Re_0pt95 .nml file
sed 's/Pt_ratio/'$Pt'/' /Users/welchla/Desktop/Dakota/efficient_global/$par_num/Re_0pt95/
Description/fun3d.temp > /Users/welchla/Desktop/Dakota/efficient_global/$par_num/Re_0pt95/Description/
fun3d.nml

# change Re_2pt96 .nml file
sed 's/Pt_ratio/'$Pt'/' /Users/welchla/Desktop/Dakota/efficient_global/$par_num/Re_2pt96/
Description/fun3d.temp > /Users/welchla/Desktop/Dakota/efficient_global/$par_num/Re_2pt96/Description/
fun3d.nml
```

Bibliography

- [1] Zikidis, K., Skondras, A., and Tokas, C., “Low Observable Principles, Stealth Aircraft and Anti-Stealth Technologies,” *Journal of Computations & Modelling*, Vol. 4, 2014, pp. 137–139.
- [2] Raymer, D. P., *Aircraft Design: A Conceptual Approach*, American Institute of Aeronautics and Astronautics, Reston, VA, 5th ed., 2012.
- [3] Oberkampf, W. L. and Roy, C. J., *Verification and Validation in Scientific Computing*, Cambridge University Press, New York, 1st ed., 2010.
- [4] Radaspiel, R., Burnazzi, M., and Scholz, P., “Active flow control for high lift with steady blowing,” *The Aeronautical Journal*, Vol. 120, 2016, pp. 171–200.
- [5] Cattafesta, L. N. and Sheplak, M., “Actuators for Active Flow Control,” *Annual Review of Fluid Mechanics*, Vol. 43, 2011, pp. 247–272.
- [6] Nelson, R. C., Corke, T. C., and Othman, H., “A Smart Wind Turbine Blade Using Distributed Plasma Actuators for Improved Performance,” *Aerospace Sciences Meeting*, Vol. 46, 2008.
- [7] Huang, L., Huang, P., and LeBeau, R., “Numerical Study of Blowing and Suction Control Mechanism on NACA 0012 Airfoil,” *Journal of Aircraft*, Vol. 2, 2012, pp. 363–381.
- [8] Pehlivanoglu, Y. V., Yagiz, B., Kandil, O., and Baysal, O., “Particle Swarm Optimization of Suction and Blowing on Airfoils at Transonic Speed,” *Journal of Aircraft*, Vol. 47, 2010, pp. 1955–1965.

- [9] Smith, D. W. and Walker, J. H., “Tests of an Area Suction Flap on an NACA 64A010 Airfoil at High Subsonic Speeds,” *NASA Technical Note*, Vol. D-310, 1960.
- [10] Qin, N., Zhu, Y., and Shaw, S., “Numerical study of active flow control for transonic aerodynamics,” *International Journal of Numerical Methods for Heat and Fluid Flow*, Vol. 14, 2004.
- [11] Seifert, A., David, S., Fono, I., Stalnov, O., and Dayan, I., “Roll Control via Active Flow Control: From Concept to Flight,” *Journal of Aircraft*, Vol. 47, 2010.
- [12] Jiyuan Tu, Guan-Heng Yeoh, C. L., *Computational Fluid Dynamics A Practical Approach*, Elsevier, Waltham, MA, 2nd ed., 2013.
- [13] Wilcox, D. C., *Turbulence Modeling for CFD*, DCW Industries, 5354 Palm Drive, La Caada, CA, 3rd ed., 2006.
- [14] Arora, J. S., *Introduction to Optimum Design*, Elsevier, Waltham, MA, 3rd ed., 2012.
- [15] Sandia National Laboratories, *Dakota, A Multilevel Parallel Object-Oriented Framework for Design Optimization, Parameter Estimation, Uncertainty Quantification, and Sensitivity Analysis: Version 6.3 Users Manual*, 2014.
- [16] Biedron, R. T., Derlaga, J. M., Gnoffo, P. A., Hammond, D. P., Jones, W. T., Kleb, B., Lee-Rausch, E. M., Nielsen, E. J., Park, M. A., Rumsey, C. L., Thomas, J. L., and Wood, W. A., *FUN3D Manual: 12.4*.
- [17] Freeman, J. A. and Roy, C. J., “Global Optimization Under Uncertainty for Tractor-Trailer Base Flaps,” *ASME Journal of Verification, Validation, and Uncertainty Quantification*, Vol. 1, 2016.

- [18] Ferson, S., Oberkampf, W., and Ginzburg, L., “Model Validation and Predictive Capability for Thermal Challenge Problem,” *Computer Methods in Applied Mechanics and Engineering*, Vol. 197, 2008.
- [19] Roy, C. S. and Balch, M. S., “A Holistic Approach to Uncertainty Quantification with Application to Supersonic Nozzle Thrust,” *International Journal for Uncertainty Quantification*, Vol. 41, 2004, pp. 1005–1013.
- [20] Scott, J., “Area Rule and Transonic Flight,” 2002, [<http://www.aerospaceweb.org/question/aerodynamics/q0104.shtml>; accessed 22-September-2016].
- [21] Anderson, J. D., *Fundamentals of Aerodynamics*, McGraw Hill Education, New Delhi, 5th ed., 2010.
- [22] Dannenberg, R. E. and Weiberg, J., “Section Characteristics of a 10.5-Percent Thick Airfoil with Area Suction as Affected by Distribution of Permeability,” *NASA TN 2847*, 1952.
- [23] Englar, R., Jones, G., Allan, B., and Lin, J., “2-D Circulation Airfoil Benchmark Experiments Intended for CFD Code Validation,” *AIAA Aerospace Sciences Meeting Including The New Horizons Forum and Aerospace Exposition*, Vol. 47, 2009.
- [24] Geleyns, D., *Alternative Flight Control in Civil Aviation*, Master’s thesis, Delft University of Technology, Mekelweg 2, 2628 CD Delft, Netherlands, 2016.
- [25] Nelson, R. C., *Flight Stability and Automatic Control*, McGraw Hill Education, New Dehli, 2nd ed., 1998.
- [26] “HPC Centers - Unclassified Systems,” 2013, [<https://centers.hpc.mil/systems/unclassified.html>; accessed 08-February-2017].

- [27] Rumsey, C., “Langley Research Center Turbulence Modeling Resource,” 2017,
[<https://turbmodels.larc.nasa.gov>; accessed 08-February-2017].

REPORT DOCUMENTATION PAGE				<i>Form Approved OMB No. 0704-0188</i>	
The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing the burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.					
PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.					
1. REPORT DATE (DD-MM-YYYY)		2. REPORT TYPE		3. DATES COVERED (From - To)	
4. TITLE AND SUBTITLE				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S)				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)				8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)				10. SPONSOR/MONITOR'S ACRONYM(S)	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT					
13. SUPPLEMENTARY NOTES					
14. ABSTRACT					
15. SUBJECT TERMS					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			19b. TELEPHONE NUMBER (Include area code)