McKelvey School of Engineering Theses & Dissertations

McKelvey School of Engineering

Summer 8-11-2016

Shape Optimization of Supersonic Bodies to Reduce Sonic Boom Signature

Junhui Li
Washington University in St Louis

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

Part of the Aerodynamics and Fluid Mechanics Commons

WASHINGTON UNIVERSITY IN ST. LOUIS

School of Engineering and Applied Science

Department of Mechanical Engineering and Material Science

Thesis Examination Committee:

Ramesh Agarwal, Chair

David Peters

Swami Karunamoorthy

SHAPE OPTIMIZATION OF SUPERSONIC BODIES TO REDUCE SONIC BOOM SIGNATURE

by

Junhui Li

A thesis presented to the School of Engineering and Applied Science

of Washington University in St. Louis in partial fulfillment of

the requirements for the degree of

Master of Science

Aug 2016

Saint Louis, Missouri

Table of Contents

List of Figures

List of Tables

Acknowledgements

I want to dedicate this thesis to my parents (Li Wang and Tao Li) and grandparents (Sulan Han and

Peikun Wang) for their life-time guidance, love and support.

Abstract of

Shape Optimization of Supersonic Bodies to Reduce Sonic Boom Signature

Junhui Li

Department of Mechanical Engineering and Materials Science

Washington University in St. Louis, St. Louis, MO 63130

In recent years there has been resurgence of interest by aerospace industry and NASA in supersonic transport aircraft. In recent studies, the emphasis has been on shape optimization of supersonic plane to reduce the acoustic signature of sonic boom resulting from the supersonic aircraft at high altitude in cruise flight. Because of the limitations of in-flight testing and cost of laboratory scale testing, CFD technology provides an attractive alternative to aid in the design and optimization of supersonic vehicles. In last decade, the predictive capability of CFD technology has significantly improved because of substantial increase in computational power, which allows for treatment of more complex geometries with larger meshes, better numerical algorithms and improved turbulence models for Reynolds-averaged Navier-Stokes (RANS) to reduce the predictive error. As computational power continues to increase, numerical optimization techniques have been combined with CFD to further aid in the design process.

In this thesis, two cases from the recent AIAA Sonic Boom Prediction Workshop have been simulated and one of them is optimized to reduce the sonic boom signature. The AIAA Sonic Boom Prediction Workshop has three models for the study of predicting sonic boom signatures and sonic boom propagation; in this thesis the Lockheed SEEB-ALR and 69 Degree Delta Wing-Body models are considered. The grid generation is conducted by ANSYS ICEM. Flow calculations are performed with ANSYS FLUENT using the compressible SA model. Excellent agreement between

the computed pressure distributions and experimental results at all positions of the models is obtained. Shape optimization of the SEEB-ALR axisymmetric body to minimize the sonic boom signature is then performed using a genetic algorithm (GA). The optimized shape shows decrease in the strength of the sonic boom signature. The results presented in thesis demonstrate that CFD can be accurately and effectively employed for shape optimization of a supersonic airplane to minimize the boom signature.

Chapter 1

Introduction

This chapter provides the background of the Computational Fluid Dynamics application of flow field simulation of supersonic aircraft and the motivation behind this research on shape optimization of these components. It is surmised that this research should eventually lead to the flow field simulation and optimization of a complete supersonic aircraft to reduce its sonic boom signature. The scope of this thesis is also included.

1.1 Motivation

In recent decades, simulation-based design methodologies have been widely in aerospace and other industries. Computer-aided design (CAD) techniques are used to accurately describe the physical model in digital format that is used for aerodynamics, structural and other analysis by solving the appropriate partial differential equations using the numerical algorithm on a computer. Computational fluid dynamics (CFD) technology is one of the technologies that are widely used for flow field simulations and analysis in a wide range of industrial applications. Thus it is possible now for the engineers to more precisely predict the performance of many industrial products.

In recent years there has been resurgence of interest in aerospace industry and NASA in supersonic transport aircraft. The emphasis has been on shape optimization of supersonic plane to reduce the acoustic signature of sonic boom resulting from the supersonic aircraft at high altitude in cruise flight. Because of the limitations on physical testing, CFD technology provides an attractive

alternative to aid in the aerodynamic design of future supersonic transport vehicles. In last couple of decades, there have been significant developments in the solution of Compressible Reynolds-averaged Navier-Stokes(RANS) equations for complex geometries because of improvements in mesh generation software, numerical algorithms and turbulence models aided by an enormous increase in computational power.

The AIAA has estimated a Sonic Boom Prediction Workshop website [1] which describes three benchmark models for the study of sonic boom signatures and sonic boom propagation. These three models can be used by the researchers to participate in the workshop and compare the result of their simulations as well as share their experience. The experimental data and flow information are provided; thus the simulation can be validated. The measured pressure disturbance in near-field is the sonic boom signatures, which can also be used to study the boom propagation to the ground (which can also be studies by CFD). To minimize the boom signature, shape optimization codes can be used.

1.2 Brief Review of Literature

To address the study of supersonic configurations with low sonic boom, NASA [2] has supported three major tests which have been conducted at the NASA Ames Research Center (ARC). For the first test LM1, although an improvement was made by distortion correction, the experimental data of SEEB-Aft Lift Relaxation axisymmetric model and CFD showed a 2/3 variation in results. For the second test LM2, the incremental results showed how CFD should be applied to achieve a good match. The LM3 improved the measurement accuracy and matched CFD prediction with good precision.

Maute etc. [3] proposed a computational method which combines a CFD method for determining the near-field pressure field and an acoustic scheme for predicting the initial shock pressure rise on the ground. This method also featured an adjoint method for accelerating the constrained shape optimization process and illustrated it with sonic boom analysis of an F5 fighter aircraft and the optimization of its nose section. Chan[4] developed a multi-objective genetic algorithm for shape optimizing a supersonic aircraft, in which both the sonic boom and drag were optimized simultaneously. Two supersonic business jets were shape optimized by this method and the loudness and peak overpressure of the results from the two aircrafts were compared.

1.3 Scope of the Thesis

In the supersonic flow part of this thesis a Lockheed SEEB-ALR and a 69 Degree Delta Wing-Body was considered for CFD simulation. Turbulent viscous flow is computed employing the compressible Reynolds-Average Navier-Stokes (RANS) equation embedded with the Spalart-Allmaras (SA) turbulence model. The simulations are validated by comparing the results with experimental data. For shape optimization of the SEEB-ALR model to minimize its sonic boom signature, a Single Objective Genetic Algorithm (SOGA) is employed.

Chapter 2

Methodology

In this chapter, the methodology used for flow field calculation and optimization is briefly described. Some relevant aspects of CFD simulation methodology are discussed. The turbulence model and boundary layer setting is provided. Genetic Algorithm is applied for optimization. The theory behind the Genetic algorithm is briefly described.

2.1 Methodology of Geometry Modeling and Mesh Generation

The two geometries considered are Lockheed SEEB-ALR and 69 Degree Delta Wing-Body. These models are provided by the AIAA Sonic boom Prediction Workshop. The models are rebuilt by us using CATIA V5. CATIA V5 is a well-known Computer Aided Design software which is popular in aerospace applications. The details of these two models are presented in Chapter 3.

The volume mesh around these two models is hexahedral structured type built by the software ICEM CFD. ICEM CFD has excellent capability for structured mesh generation. It is extensively used in aerospace applications. The structured hexahedral mesh provides enough quality for accurate prediction of sonic boom. The details of mesh are presented in Chapter 3.

2.2 **Methodology of CFD Simulation**

2.2.1 Brief Outline of the Numerical Modeling

The commercial CFD solver ANSYS FLUENT is employed for flow simulations. The turbulent flow field is modeled using the steady Reynolds-Averaged Navier-Stokes (RANS) equations in conjunction with the widely used turbulence model, Spalart-Allmaras (SA) turbulence model. The Roe scheme is applied to numerically solve the flow equations. Because of the supersonic far-field condition, the density-based solver in FLUENT is chosen to account for the compressibility effects. The air is assumed to obey the ideal-gas. The viscosity of air is calculated from the Reynolds number used in the experiment. The properties of the air used are listed in Table 2.1.

Table 2.1: The properties of air in the flow field

Properties	SEEB-ALR	69 Degree Delta Wingbody
Density	1.176674 kg/m^3 (Ideal gas model)	1.176674 kg/m^3 (Ideal gas model)
Cp(Specific Heat)	1006.43 J/kg-K	1006.43 J/kg-K
Thermal conductivity	0.0242 W/m-k	0.0242 W/m-k
Viscosity	4.381771*10^-5 kg/m-s	5.005933*10^-5 kg/m-s
Molecular weight	28.966 kg/kgmol	28.966 kg/kgmol

2.2.2 Turbulence Model

Spalart-Allmaras (SA) model is used to solve the Reynold-Average Navier-Stokes (RANS) equations. This model has excellent solution convergence property compared to many other models. Because of the presence of a very small flat plane (blunt area) in front of SEEB-ALR and 69 Degree Delta Wing-body, the stagnation point occurs at the center of the blunt flat area. Therefore, compared to the inviscid Euler model, RANS equations with SA model can provide much more stable solution in this area due to the effect of viscosity.

The "Standard" Spalart-Allmaras [5] One-Equation Model (SA) is given by the following equation:

$$\frac{\partial \hat{v}}{\partial t} + u_j \frac{\partial \hat{v}}{\partial x_j} = c_{b1}(1 - f_{t2})\hat{S}\hat{v} - \left[c_{w1}f_w - \frac{c_{b1}}{\kappa^2}f_{t2}\right]\left(\frac{\hat{v}}{d}\right)^2 + \frac{1}{\sigma}\left[\frac{\partial}{\partial x_j}\left((v + \hat{v})\frac{\partial \hat{v}}{\partial x_j}\right) + c_{b2}\frac{\partial \hat{v}}{\partial x_i}\frac{\partial \hat{v}}{\partial x_i}\right]$$

where $\hat{v}$ is the modified turbulent viscosity. The turbulent eddy viscosity is computed from:

$$\mu_t = \rho\hat{v}f_{v1}$$

where

$$f_{v1} = \frac{\chi^3}{\chi^3 + c_{v1}^3}$$

and

$$\chi = \frac{\hat{v}}{v},$$

ρ is the density, $v = \mu/\rho$ is the molecular kinematic viscosity, and μ is the molecular dynamic viscosity. Additional definitions are given by the following equations:

$$\hat{S} = \Omega + \frac{\hat{v}}{\kappa^2 d^2}f_{v2}$$

where $\Omega = \sqrt{2W_{ij}W_{ij}}$ is the magnitude of the vorticity, d is the distance from the field point to the nearest wall, and

$$f_{v2} = 1 - \frac{\chi}{1 + \chi f_{v1}}, \qquad f_\omega = g[\frac{1 + c_{\omega 3}^6}{g^6 + c_{\omega 3}^6}]^{1/6}$$

$$g = r + c_{\omega 2}(r^6 - r)$$

$$r = min[\frac{\hat{v}}{\hat{S}\kappa^2 d^2}, 10]$$

$$f_{t2} = c_{t3}exp(-c_{t4}\chi^2)$$

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

The boundary conditions are:

$$\hat{v}_{wall} = 0, \qquad \hat{v}_{farfield} = 3v_\infty : to : 5v_\infty$$

Note that these boundary conditions on the SA turbulence field variable $\hat{v}$ correspond to turbulent kinematic viscosity values of:

$$v_{t,wall} = 0, \qquad v_{t,farfield} = 0.210438v_\infty : to : 1.294234v_\infty$$

The constants are:

$$c_{b1} = 0.1355, \qquad \sigma = \frac{2}{3}, \qquad c_{b2} = 0.622, \qquad \kappa = 0.41$$

$$c_{\omega 2} = 0.3, \qquad c_{\omega 3} = 2, \qquad c_{v1} = 7.1, \qquad c_{t4} = 0.5$$

$$c_{\omega 1} = \frac{c_{b1}}{\kappa^2} + \frac{1 + c_{b2}}{\sigma}$$

2.2.3 Boundary Conditions

Boundary conditions in the far-field are set as the pressure-far-field boundary condition. The boundary condition downstream the body is applied as the pressure-outlet boundary condition. The pressure is set at 101325 Pa and the temperature is set at 300K. The operating conditions are 0 Pa, with gravity neglected. Turbulence viscosity ratio in the far-field and outlet are set to be 5%. The

cross sectional plane of the body and the far-field is defined as the symmetry boundary. This symmetry condition allows halving of the computational domain.

2.2.4 Convergence Monitors and Criteria

Two convergence monitors are employed to check if the final results are convergent to the desired tolerance. The residuals of all six governing flow equations including the turbulence model equations are monitored until they achieve the absolute value of 10^{-6}; at that stage the solution is considered to be converged. A drag coefficient (Cd) monitor is also added to evaluate convergence; the solution is considered converged if Cd achieved a constant steady state value. In the real computations, it was found that the scaled residual values were not reach 10^{-6} for all six flow governing equations. The residuals oscillate around 10^{-4} and did not change with increase number of iterations. Under this circumstance, the solution was considered to be converged if the Cd monitor achieved a constant steady state value.

2.3 Overview of the Genetic Algorithm

Genetic algorithms are a class of stochastic optimization algorithms inspired by biological evolution [6]. In GA, a set of input vectors, called individuals, is iterated over, successively combining traits of the best individuals until convergence is achieved [7]. In general, a GA employs the following steps:

1.   Initialization: Randomly generate N individuals.

2.   Evaluation: Evaluate the fitness of each individual.

3.   Natural Selection: Remove a subset of the individuals (often individuals with the lowest fitness).

4.   Reproduction: Create a new generation of individuals.

a. Parent Selection: Select pairs of individuals to produce an offspring. This is usually achieved by roulette wheel sampling; the possibility of selecting some individual h_i for reproduction is given by the formula below:

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

b. Crossover: Exchange information between the parent individuals. A crossover function is then preformed to produce the offspring form the selected parents. Generally, crossover is implemented by choosing a crossover point on each individual and swapping alleles- or vector elements- at this point as illustrated in Fig 2.1.

x_1^1	x_2^1	x_3^1	...	x_{n-2}^1	x_{n-1}^1	x_n^1
x_1^2	x_2^2	x_3^2	...	x_{n-2}^2	x_{n-1}^2	x_n^2

x_1^2	x_2^2	x_3^1	...	x_{n-2}^1	x_{n-1}^1	x_n^1
x_1^1	x_2^1	x_3^2	...	x_{n-2}^2	x_{n-1}^2	x_n^2
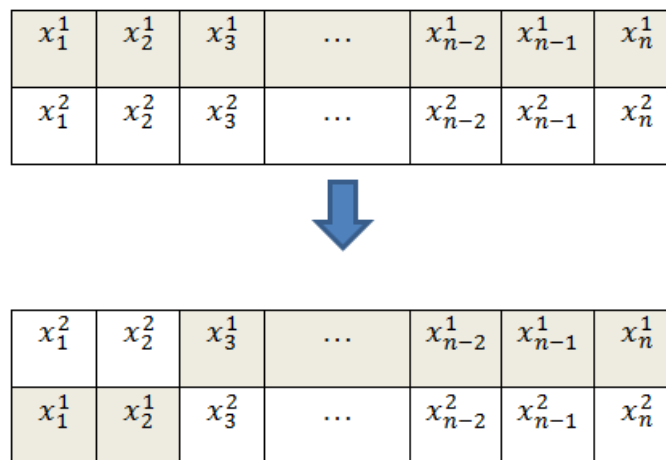
Figure 2.1: Illustration of the general Crossover function in GA.

c. Mutation: Randomly alter some small percentage of the population.

5. Check for Convergence: If the solution has converged, return the fittest individual. If the solution has not yet converged, repeat the cycle starting at step 2 with the new generation.

Convergence often occurs after a certain number of generations meeting a user defined tolerance criteria for acceptable change in objective values from one generation to next.

2.4 Single Objective Genetic Algorithm (SOGA)

Using MATLAB as the platform, Genetic Algorithm can be applied to the shape optimization of SEEB-ALR. Curve generations, ICEM meshing and FLUENT flow field calculations are performed in each generation of SOGA with a number of randomly generated curves and are continued to a number of generations until the solution converges to an optimal shape to achieve the objective value, and repeated as shown in Figure 2.2.
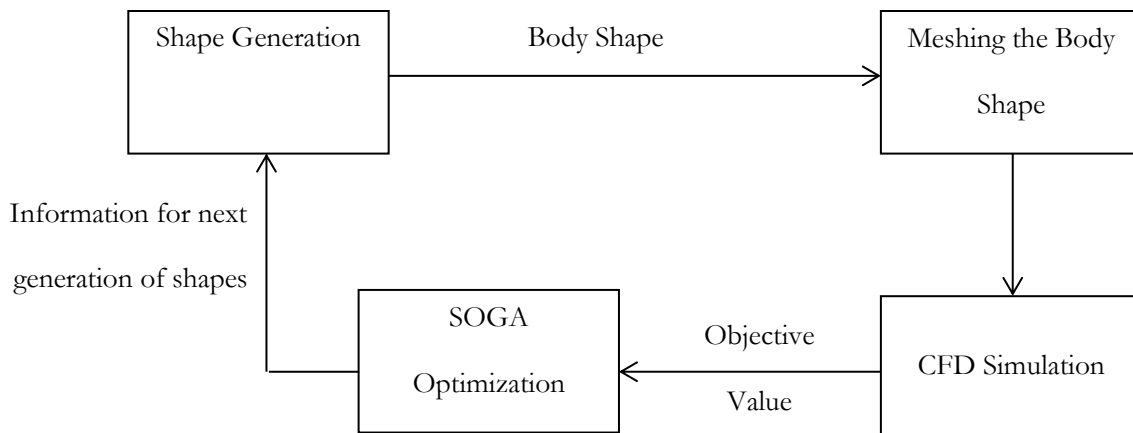
Figure 2.2: General schematic of information flow in shape optimization process.

Bezier curve polynomials are used to create the random body profiles in a given generation of SOGA. A Bezier curve is a parametric curve that is defined by the control points, whereas the curve itself generates a smooth path among the control points. They have desirable characteristics for use in a GA since the initial and final points and slopes can be fixed, control point information between the individuals is easily swapped, and changing a control point results in a similar change to the

curve itself [8]. Figure 2.3 shows a set, or generation, of three random curves. These curves represent the randomly generated shapes. The green dots represent the control points, whereas the red line represents the resulting curve.
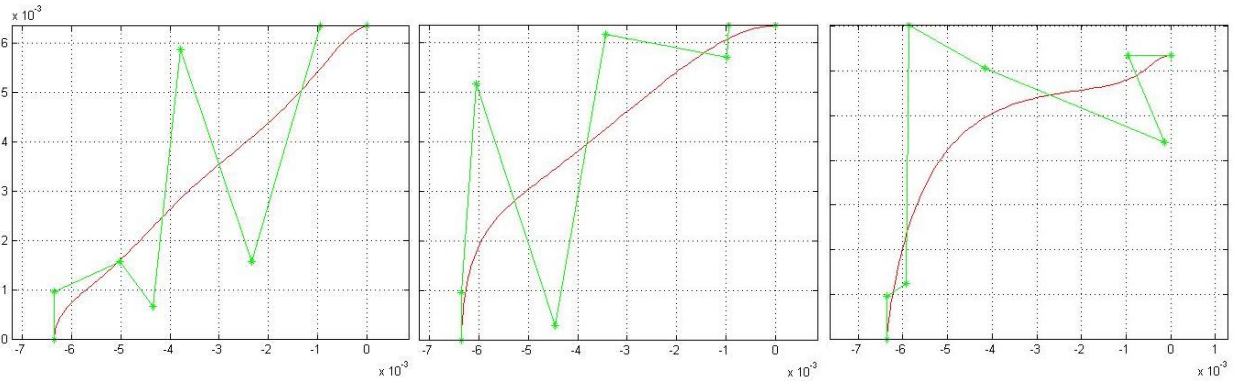
Figure 2.3: Three randomly generated curves and their Bezier control points.

As an example, the three curves in Figure 2.3 are used to generate the geometries and meshes for three body shapes in a typical generation of SOGA. The flow field is then calculated using FLUNET. The three steps, namely the generation of body shapes, their meshing by ICEM and flow field solution by FLUENT, are repeated until all the needed information from all individuals (body shapes) in a generation is recorded. Then the sonic boom signatures for all the body shapes in a generations are recorded and loaded into the SOGA, which runs evaluation, natural selection, and reproduction functions on the collected data. The objective function is the minimization of maximum pressure in the sonic boom. In the optimization process, the objective function information for each body. For the next generation by following the steps listed below:

(a) Evaluation and Natural Selection

The objective values for each curve (body shape) are analyzed and compared to each other. In our case, minimization of the objective value is desired. Each objective value is sorted from best to worst, where the lowest value of pressure in the sonic boom signature is the best and highest value is

11

the worst. Using the criteria, the fittest half of the data "survives" and is retained and used to reproduce and influence the next generation in SOGA.

(b) Reproduction and Mutation

Reproduction involves parent selection, crossover, and mutation. From the remaining half data, from previous generation pairs of parent shapes are selected using a skewed random draw; the more fit a function is, higher is its probability of being selected. The paired curves then undergo crossover, where they exchange Bezier point information. The exchange can be a complete swap of certain points or a weighted average of the characteristics of certain points. Both type of crossover and type of exchange are selected at random. This process is repeated until the desired number of daughter shapes is created and a new generation is formed. The final step is mutation. Randomly selected Bezier points of randomly selected curves are randomly altered, adding more diversity into the generation to increase the range of body shapes.

(c) Solution Convergence

Finally, the SOGA checks if the curves obtained in a given generation satisfy the convergence criteria. The convergence criteria is based on whether the body shapes in a generation are close to each other and the difference in the objective values in that generation is within a specified tolerance. If the changes in the body shape and in the objective values in a few subsequent generations (usually 2 to 3) are in acceptable range, the solution is considered converged. Once the solution converges, SOGA gives the fittest (almost optimal) body shape meeting the desired objective. If the solution has not yet converged, the new generation is loaded into the shape generating function and the process is repeated.

Chapter 3

Numerical Solution and Validation of SEEB-ALR and 69 Degree Delta Wing-body

This chapter describes the numerical solution and its validation for the flow field of SEEB-ALR and 69 Degree Delta Wing-body. The geometries and meshes of these two models are built by ICEM. The flow field simulations performed by FLUENT and are compared with the experimental data to validate the accuracy of the CFD simulations for sonic boom prediction.

3.1 Validation Test Cases

3.1.1 SEEB-ALR

The SEEB-ALR is an axisymmetric model shown in Figure 3.1. The reference length of the model is 17.68 inches [1]. The model is placed in a wind tunnel flow condition where the free stream Mach number is 1.6 at an angle of attack of 0°. The Reynolds number based on the freestream velocity is $Re = 6.42 \times 10^6$.
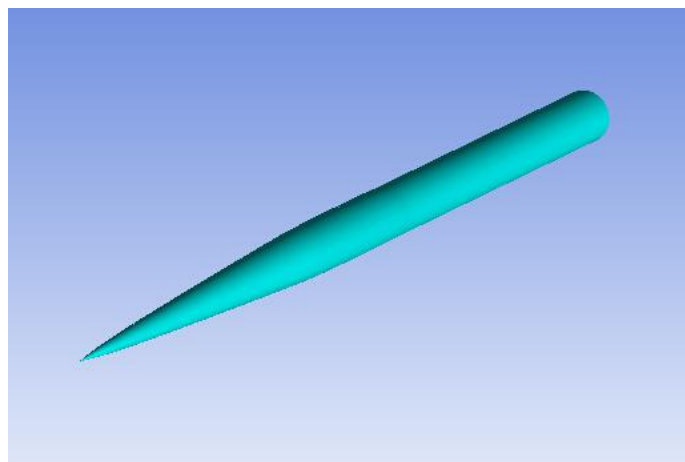
Figure 3.1: SEEB-ALR geometry

3.1.2 69° Delta Wing-Body

The 69° Delta Wing-Body is a symmetric body shown in Figure 3.2 and Figure 3.3. The wing-body model and support structure are both included in the simulation. The reference length of the model is 6.9 inches. The model is placed in a wind tunnel condition where the freestream Mach number is 1.7 at an angle of attack of 0° and the Reynolds number based on the freestream velocity is $Re = 2.43 \times 10^6$.
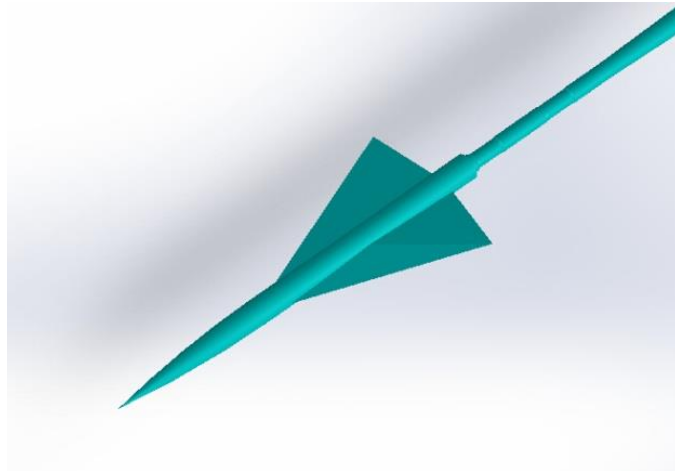
Figure 3.2: 69° Delta Wing-Body
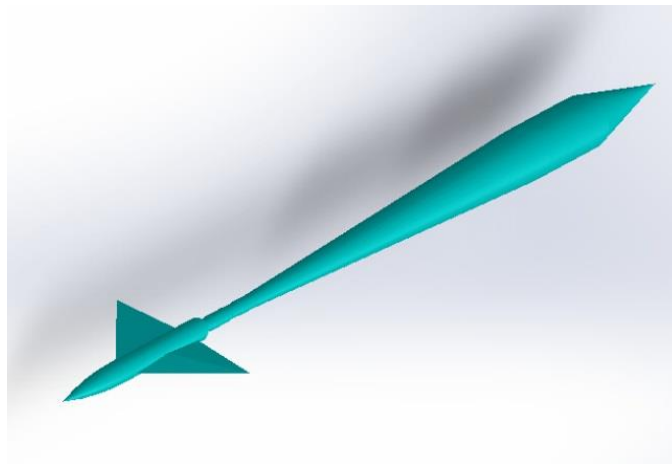
Figure 3.3: 69° Delta Wing-Body with support structure

3.2 **Mesh Generation**

3.2.1 **SEEB-ALR**

The grid for SEEB-ALR is generated by ICEM CFD and contains 1.4 million hexahedra cells as shown in Figure 3.4. The model is axisymmetric about the centerline, therefore only half of the model is considered for CFD simulations. The far-field boundary is at a distance $1/10^{th}$ of the reference length of SEEB-ALR. The mesh is aligned with the shock wave at Mach angle of 38.68° in the radial direction. O-Block topology is applied for the mesh emanating from the far-field towards the body in the direction of propagation of shock wave. According to the geometry from AIAA Sonic Boom Workshop, the front cusp of the body is a small plane which can be meshed by hexahedra cells. The characteristic length of this plane is less than 0.1% of the reference length of the model.
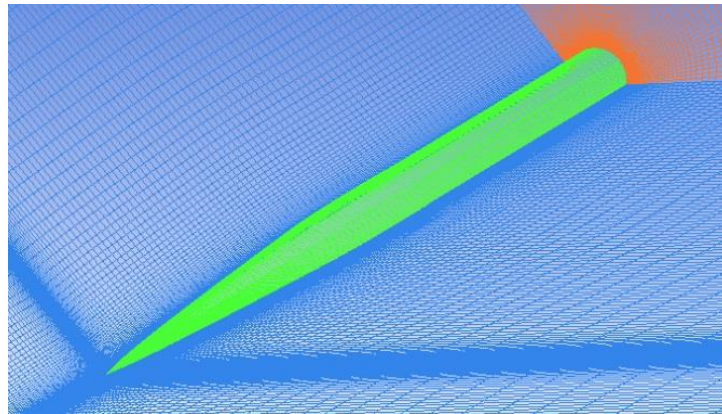
Figure 3.4: Mesh around SEEB-ALR

3.2.2 69° Delta Wing-Body

The grid for 69° Delta wing-body is generated by ICEM CFD and contains 4.57 million hexahedra cells as shown in Figure 3.5 and Figure 3.6. The model is symmetric about the center plane, therefore only half of the model is considered for CFD simulations. The far-field boundary is at a distance $1/10^{th}$ of the reference length of Delta wing-body. The mesh is aligned with the shock wave at Mach angle of 36.03 degree in the radial direction. O-Block topology is applied for the mesh emanating from the far-field towards the body in the direction of propagation of shock wave. Y-Blocks topologies are applied to build the mesh near the delta wing. For hexahedra mesh, Y-blocks enable each triangular pyramid block to split into three rectangular pyramid blocks.
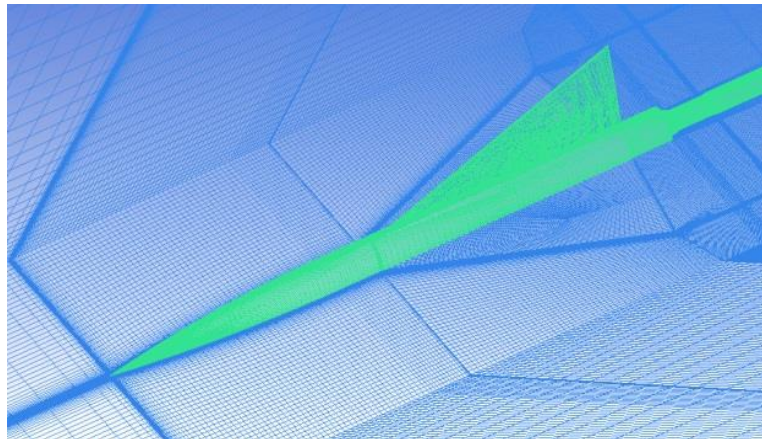
Figure 3.5: Mesh around the 69° Delta Wing-Body
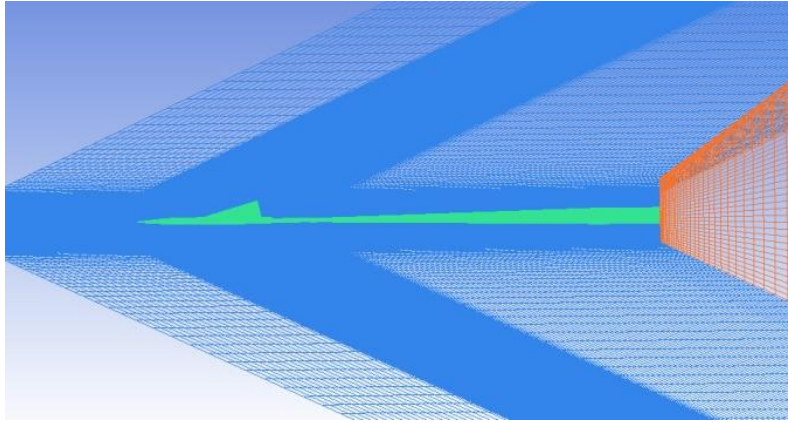
Figure 3.6: Mesh around the 69° Delta Wing-Body with support structure

3.3 Simulation Results

3.3.1 Simulation results for SEEB-ALR

As described in the AIAA Sonic Boom Workshop, near field pressure disturbance $\Delta P/P$ can be used to characterize the signature of sonic boom to represent the strength of the shock waves generate by the boom. The contours of $\Delta P/P$ for SEEB-ALR are shown in Figure 3.7. The solution clearly shows the shock waves emanating from the body. The contours show that the pressure disturbance reaches the maximum value in the front of the body and decreases along the x-direction. The minimum value is reached near the end position of the conical part of the body.
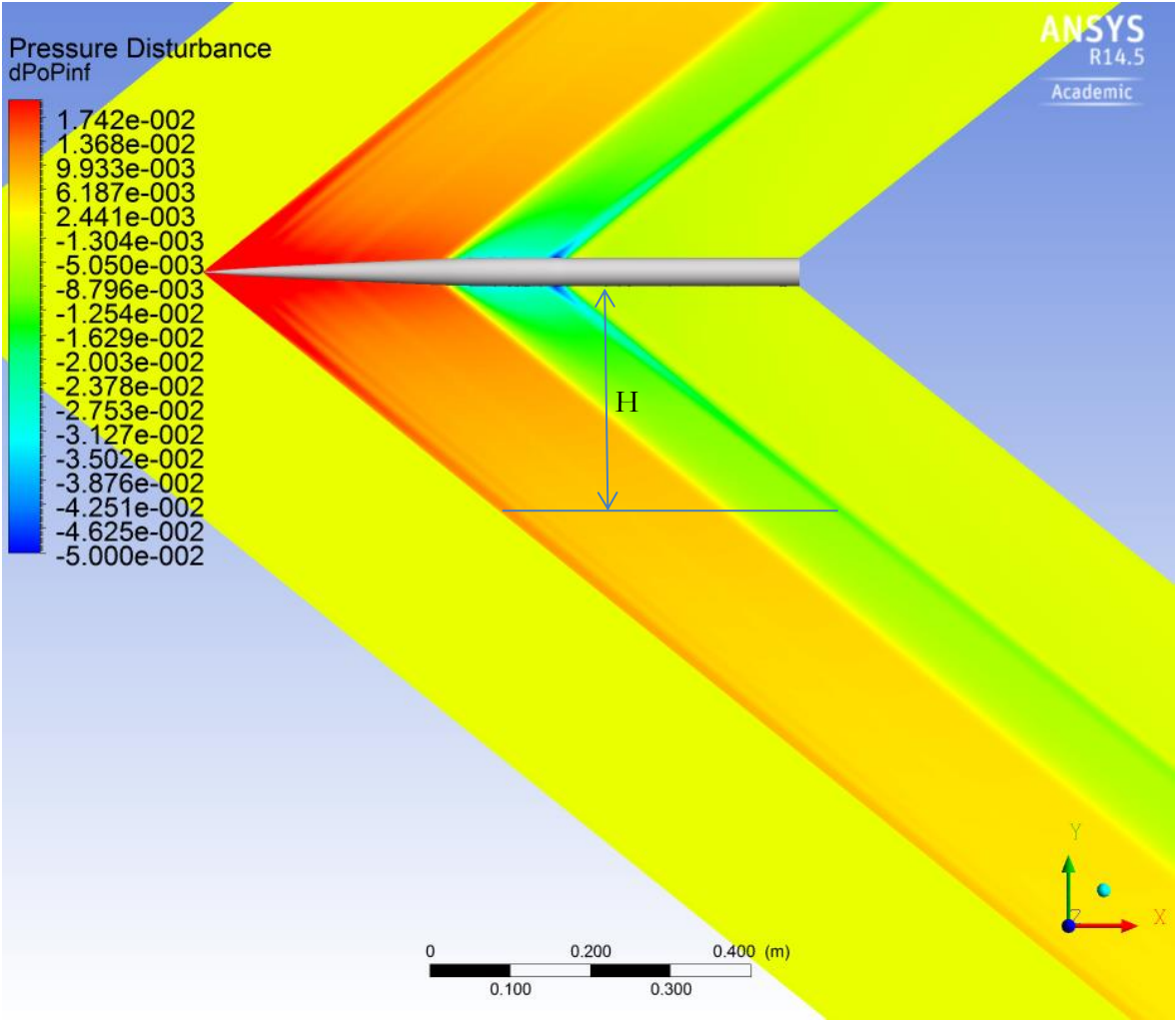
Figure 3.7: Pressure Disturbance (ΔP/P) Contours around SEEB-ALR body

The experimental data is available for the distribution of pressure disturbance along x-direction at 2 different y positions. As it shown in Figures 3.8 and 3.9 below, the pressure disturbance of the simulation result at y=21 in and y=31 are in excellent agreement with the experimental data. The simulation successfully predicts the maximum disturbance of the same magnitude at the same position as the experiment data. Thus, SEEB-ALR simulation has been successfully verified and validated; therefore the minimization of this pressure disturbance can be used as the shape optimization objective for the SEEB-ALR body.
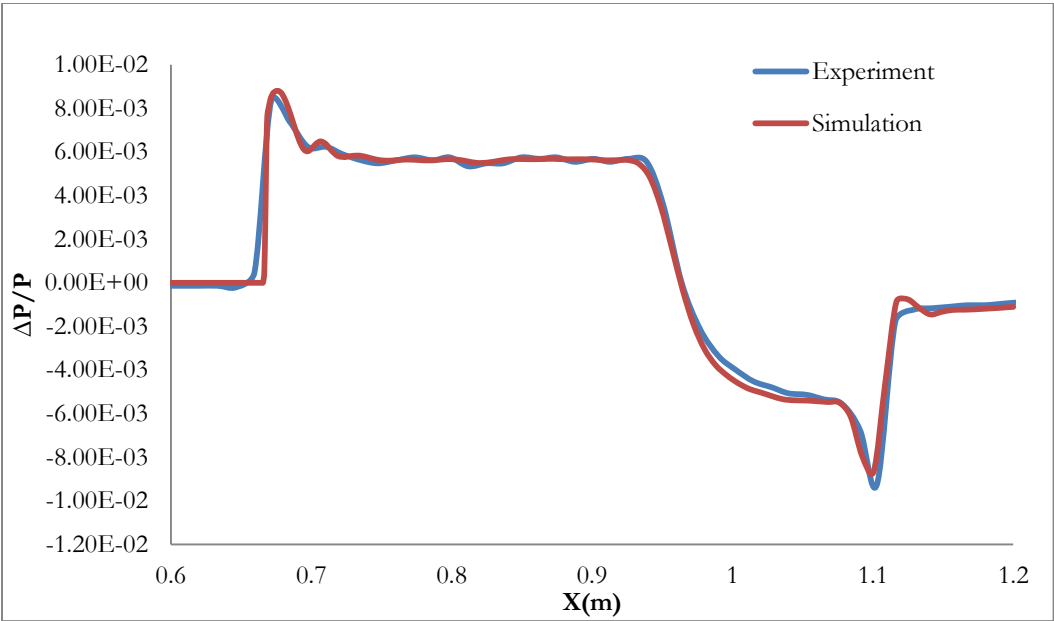
Figure 3.8: Pressure disturbance along x-direction at H=21in (H is shown in Figure 3.7)
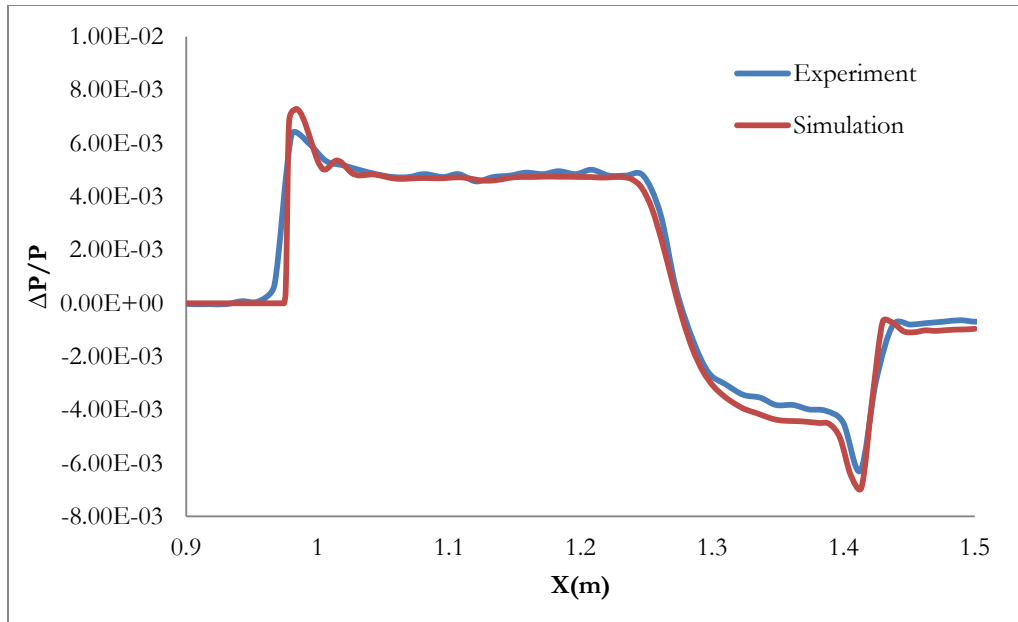
Figure 3.9: Pressure disturbance along x-direction at H=31in (H is shown in Figure 3.7)

3.3.2 Simulation results of 69° Delta Wing-Body

The contour of near field pressure disturbance ΔP/P for the 69° Delta wing-body in the x-y plane are shown in Figure 3.10. Compared to SEEB-ALR case, the shock wave pattern is much more complex. All the emanating shock waves propagate along the same direction of the Mach angle and the strength of the shock waves decrease along the direction of propagation. The shock waves occur at different locations along the x-direction. It can be noticed that the strongest shock wave occurs at the tip of the delta wing. The shock wave that occurs at the front part of the fuselage is also very strong.
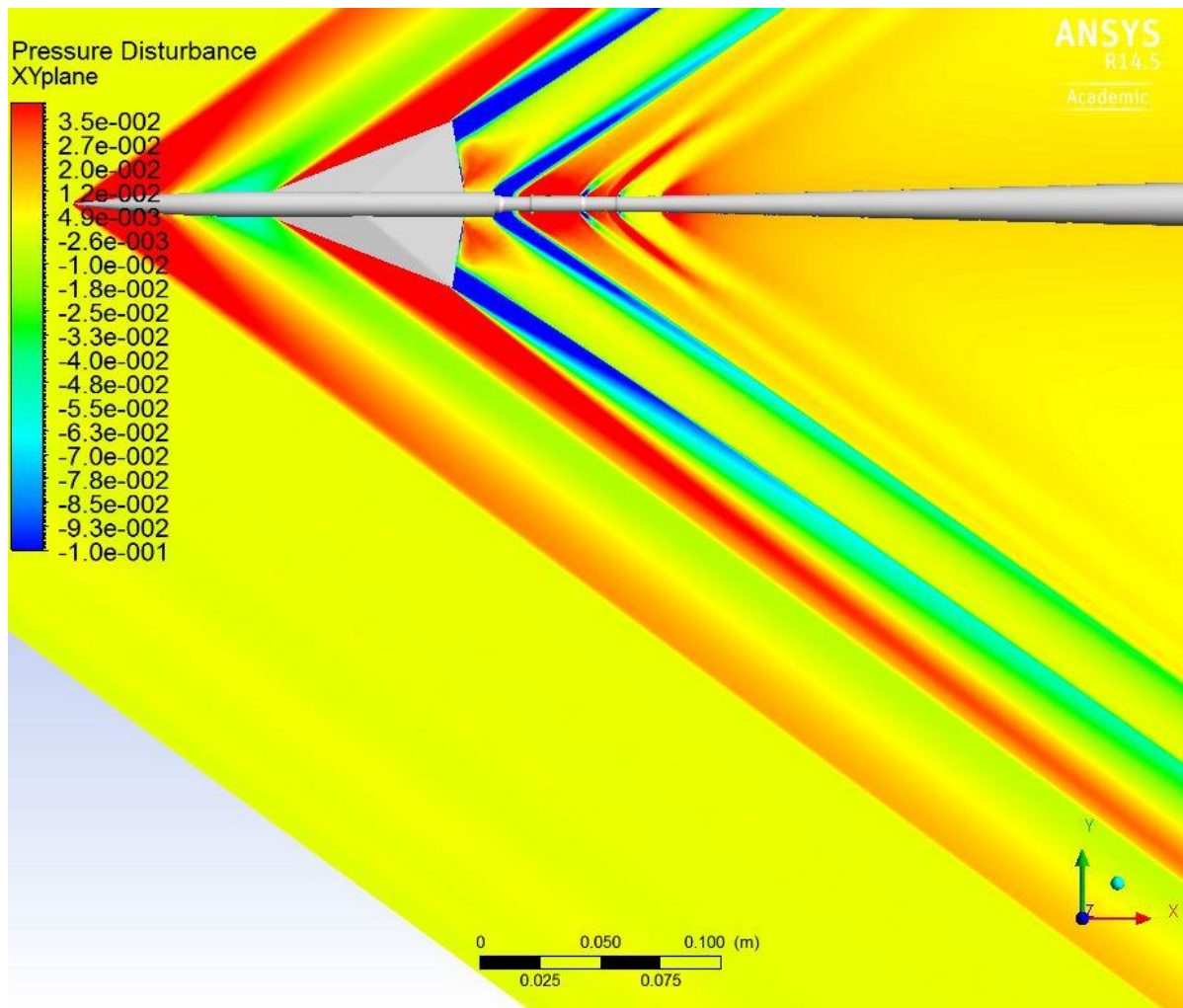
Figure 3.10: Pressure Disturbance contours in the x-y plane of the 69° Delta Wing-Body.

The contours of near field pressure disturbance $\Delta P/P$ for 69° Delta wing-body in the x-z plane are shown in Figure 3.11. This is the symmetry plane of the model and it shows significant differences in the structure of shock waves when compared to the structure in the x-y plane.
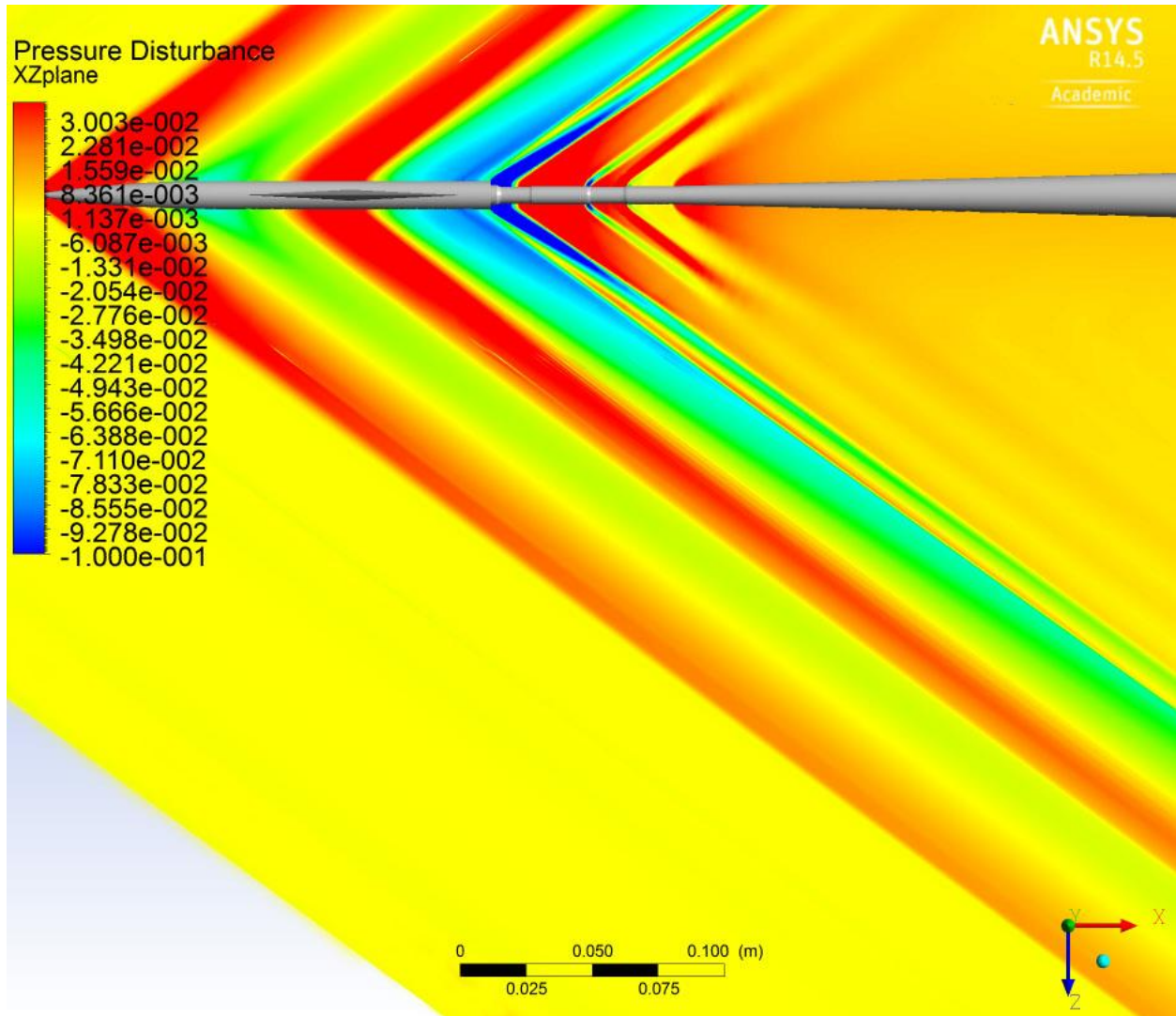
Figure 3.11: Pressure Disturbance contours in the x-z plane of 69° Delta Wing-Body.

Figures 3.12- 3.15 shows $\Delta P/P$ along the x-direction at H=25 inches (H=3.6L_{Ref}) away from the body at different azimuthal angle, φ. The simulation results show very good agreement with the

experiment at $\varphi = 0^o$ $\varphi = 30^o$ $\varphi = 60^o$ and $\varphi = 90^o$. These results shows that the Y-Block mesh provides enough mesh quality near the delta wing for accurate computations. Thus, the Delta wing-body simulation has been verified and validated; therefore the minimization of this pressure disturbance can be used as the shape optimization objective for the Delta wing-body.
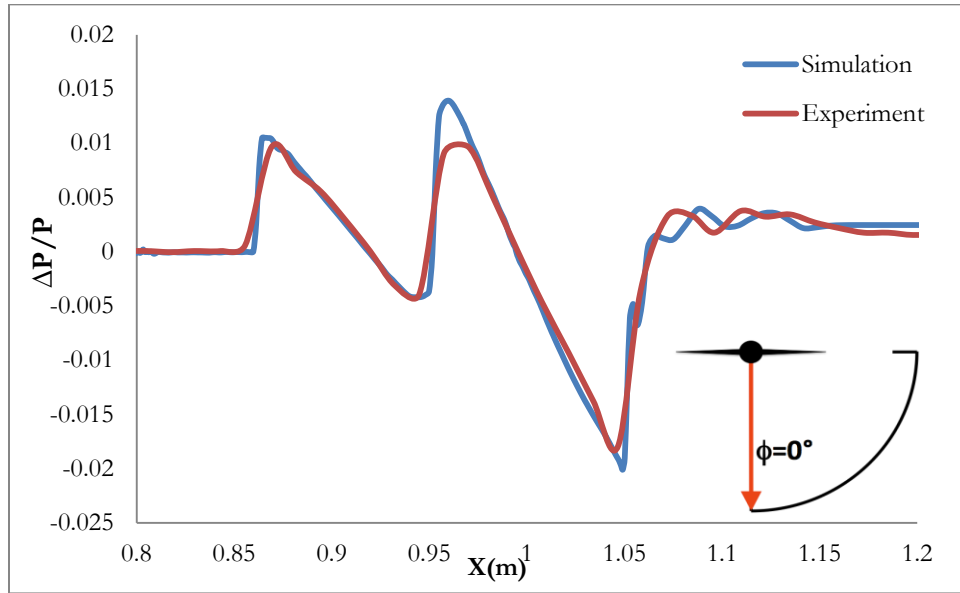
Figure 3.12: Pressure disturbance along x direction at H=25 inches at $\varphi = 0°$
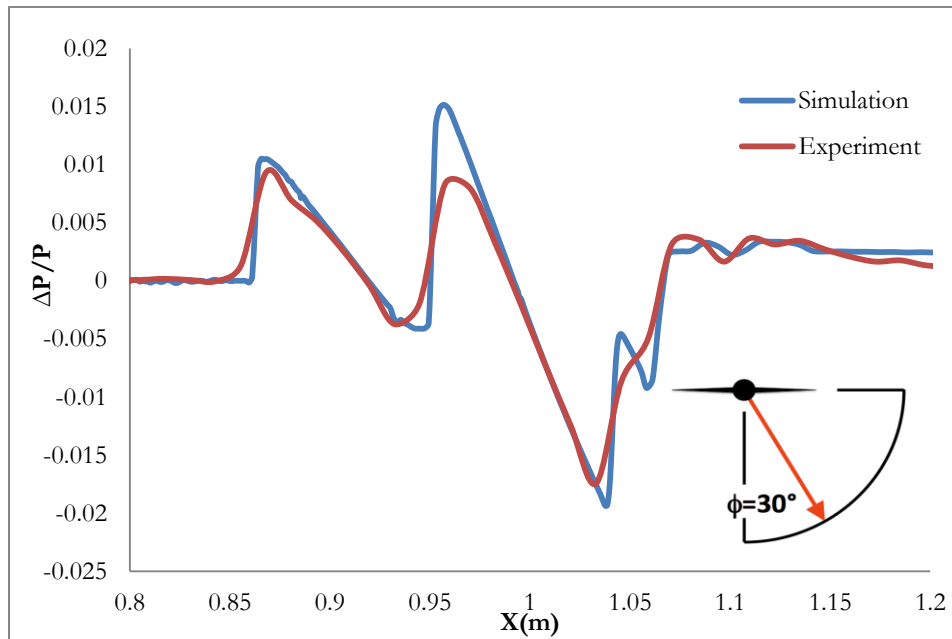
Figure 3.13: Pressure disturbance along x direction at H=25 inches at $\varphi = 30°$
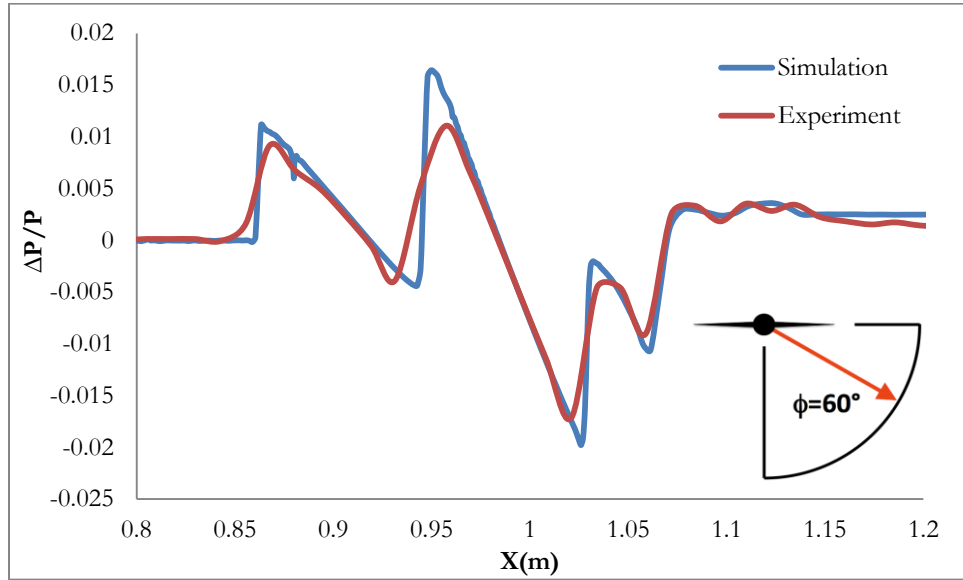
Figure 3.14: Pressure disturbance along x direction at H=25 inches at $\varphi = 60°$
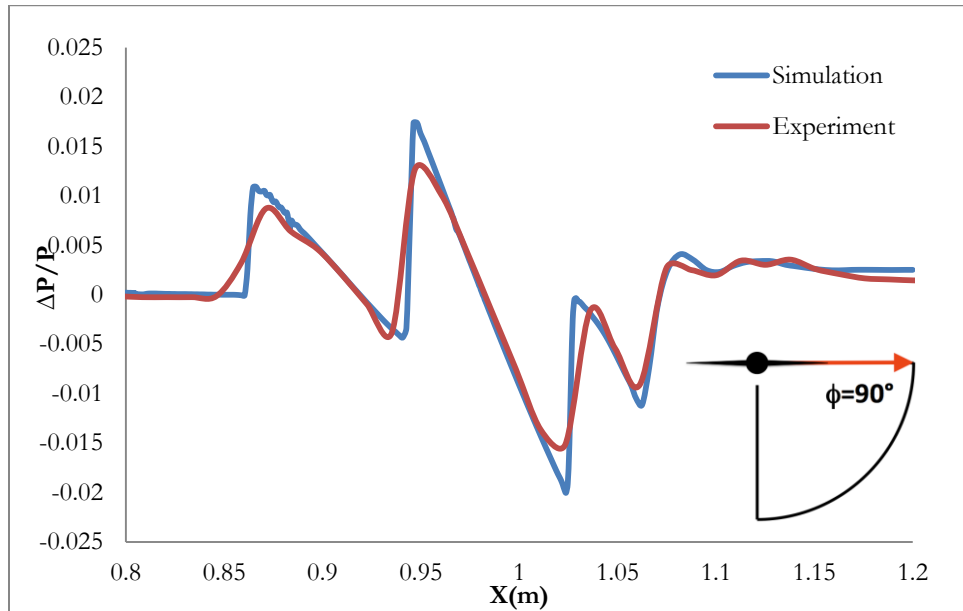
Figure 3.15: Pressure disturbance along x direction at H=25 inches at $\varphi = 90°$

Chapter 4

Shape Optimization of SEEB-ALR

In this chapter, the shape optimization procedure for SEEB-ALR body using the Genetic Algorithm (GA) and the result of optimal SEEB-ALR shape which minimizes its sonic boom signature are presented. The shapes generated using Bezier curve, the mesh generation around the shape using ICEM and the flow calculation for the shapes using FLUENT are integrated with the Genetic Algorithm to achieve the objective of shape optimization. The details of these steps are given in this chapter.

4.1 Application of Single-Objective Genetic Algorithm

A Single-Objective Genetic Algorithm (SOGA) is employed for the shape optimization of the SEEB-ALR. Minimizing the pressure disturbance is the objective of this optimization. Based on the comparison of sonic boom signature from CFD and the experimental data, it is expected that the minimized sonic boom signature will have similar pattern but with lower peak values. The absolute value of the pressure disturbance represents the strength of the sonic boom. Hence the objective or fitness function is defined as:

$$fitness = |P - Pinf| = |P - 101325|$$

All the parameters used in the application of the Genetic Algorithm for shape optimization are given in Table 4.1

Table 4.1: Genetic Algorithm parameters

GA parameters	Description

Generation Size	10 individuals per generation
Number of Generations	Maximum of 100 generations if convergence is not reached
Selection Type	Roulette Wheel Selection
Crossover Rate	0.5
Mutation Rate	0.4
Error of Mutation Constant	0.8, which determines how much mutation affects the curve as generations proceeds

Figure 4.1 shows a detailed schematic of information flow in the optimization process.
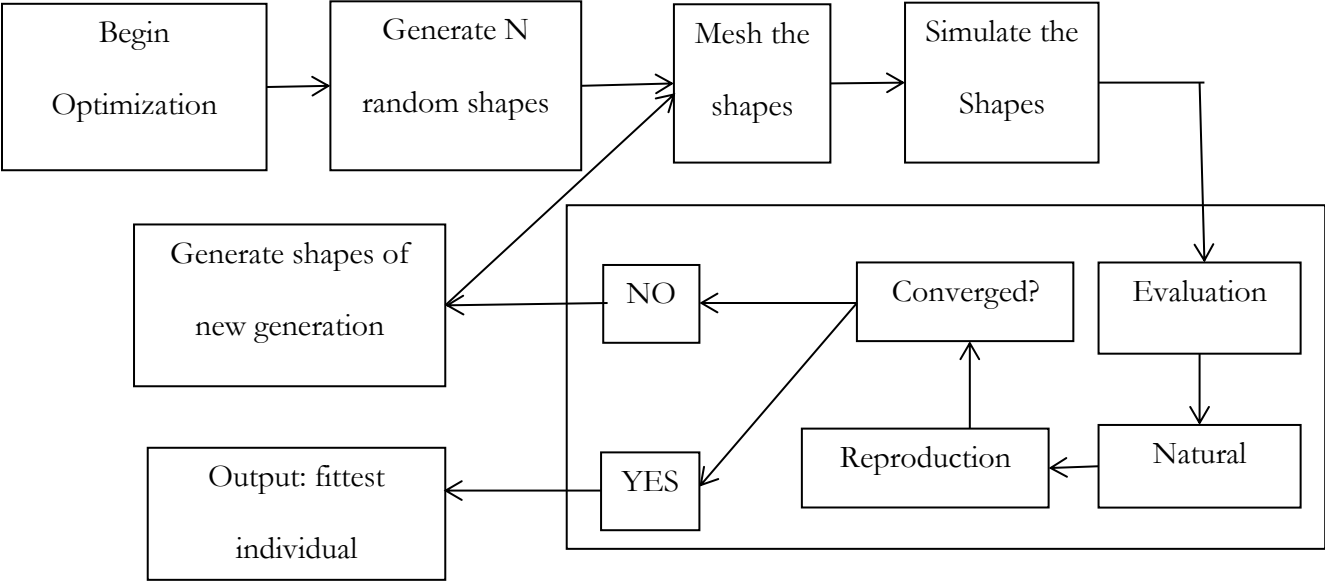
Figure 4.1: Schematic of information flow in SOGA optimization process.

4.2 **Shape Parameterization**

The shape of the SEEB-ALR model is an axisymmetric body and can be generated by rotating a 2D model. The front part of SEEB-ALR is a cone rotated about a straight line at an angle to the x-axis and the back part is a cylinder. Thus the SEEB-ALR model is a cone-cylinder. From the pressure disturbance results of SEEB-ALR in Figure 3.8 and Figure 3.9 in Chapter 3, the pressure returns back to the atmospheric pressure in the cylindrical part of the model. Thus the shape optimization is done for the conical part of the model.

The cone is generated by rotating one Bezier curve. Six control points are set to be the constraints for the random curves generation. To ensure that the starting point and the end point of the curve are fixed, the 1st and 6th control point are fixed. To ensure smooth transition from cone to cylinder, the 5th control point is also fixed. Thus there are 3 control points that are varied randomly to generate a shape. For each control point, a window which limits the region in which a control point can more is defined. The upper limit and lower limit of movement for all six control points is given in Table 4.2.

Table 4.2: Coordinate range for each control point of the Bezier curve for first round of GA

Coordinate	Upper limit	Lower limit
x_1	Fixed at x=0	
x_2	0	5 inches
x_3	5 inches	10 inches
x_4	10 inches	16.5 inches
x_5	Fixed at x=16.5 inches	
x_6	Fixed at x=17.692413 inches	

y_1	Fixed at y=0.005 inch	
y_2	0	0.5 inch
y_3	0	0.6 inch
y_4	0	0.7 inch
y_5	Fixed at y=0.69764638 inch	
y_6	Fixed at y=0.69764638 inch	

The window of each point is set to be fairly large to allow for greater possibility for change in shape. Increasing the upper limit and the lower limit of constraints in the x direction for the three randomly changed control points (with other three control points fixed) avoids possibility of generating a very unusual shape. Figure 4.2 and 4.3 show examples of two randomly generated Bezier curves.
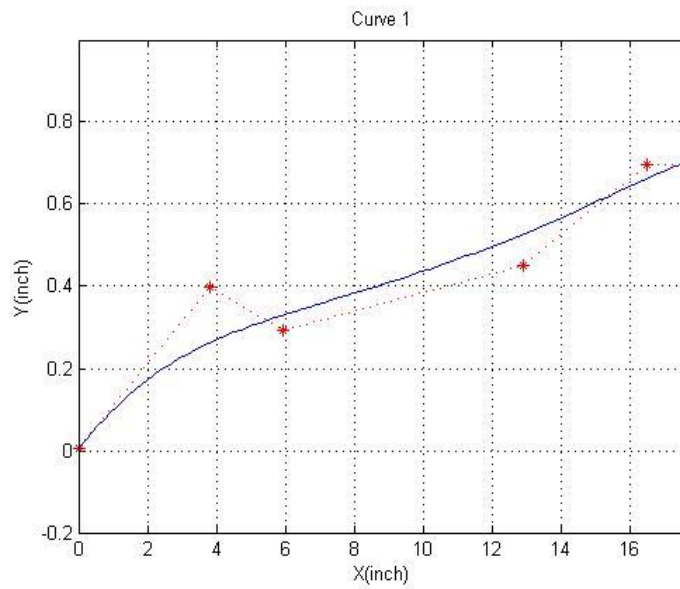
Figure 4.2: Randomly generated Bezier curve 1

28
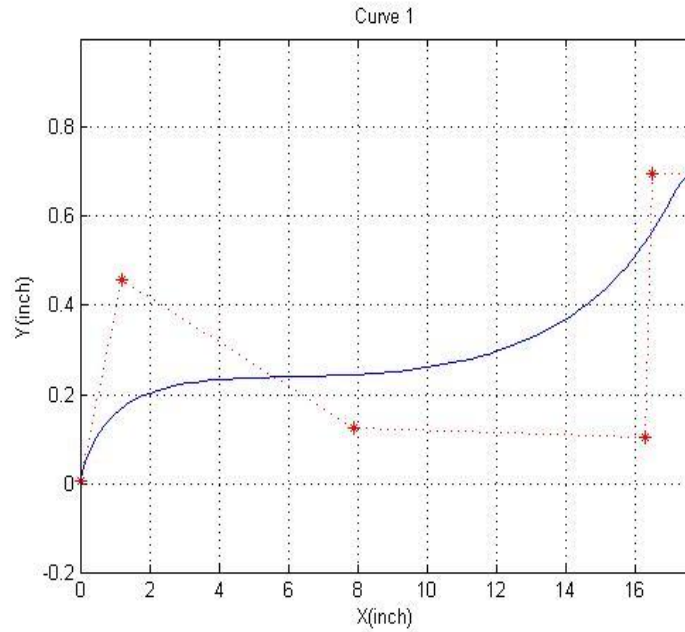
Figure 4.3: Randomly generated Bezier curve 2

4.3 **Mesh Generation**

For each random shape of SEEB-ALR need for SOGA, the grid is automatically generated by ICEM using replay file. The Bezier curve is split into 100 points and is imported to ICEM. In ICEM, the points are connected together by a curve and the curve is rotated 180 degree to form a half of the axisymmetric body (Figure 4.4). Then the fixed far-field boundary is created (Figure 4.5). The boundary in the far-field is built in the shape of a Mach cone at Mach angle of 38.68 degree to generate the mesh aligned with Mach angle for accurate shock wave capture. Thus, the computational becomes fairly small which saves the computational resources as well as creates a high density mesh in the flow field. For this cone-shape far-field, O-block method is applied to build the block. This method allows the mesh to directly converge from the far-field down to the body. The total size of the mesh is 1.4 million cells (Figure 4.6 & Figure 4.7).
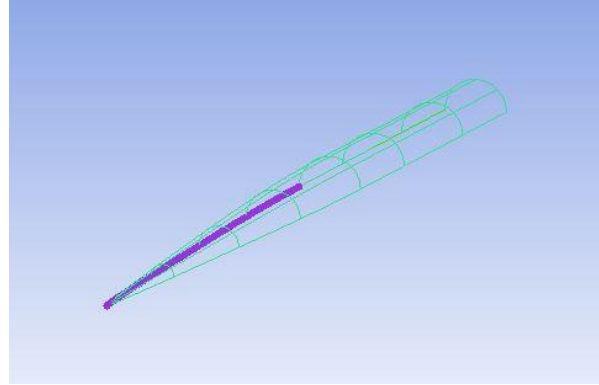
Figure 4.4: Body rotated around the curve into ICEM generated by the Bezier curve points
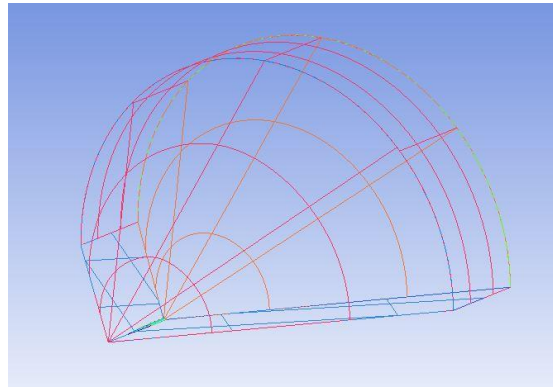
Figure 4.5: Far-field boundary surfaces created by the fixed points
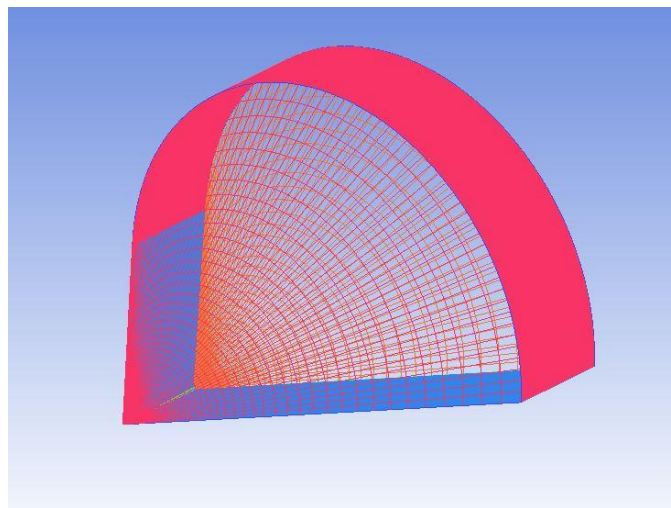
Figure 4.6: Conical mesh around a randomly generated SEEB-ALR body shape
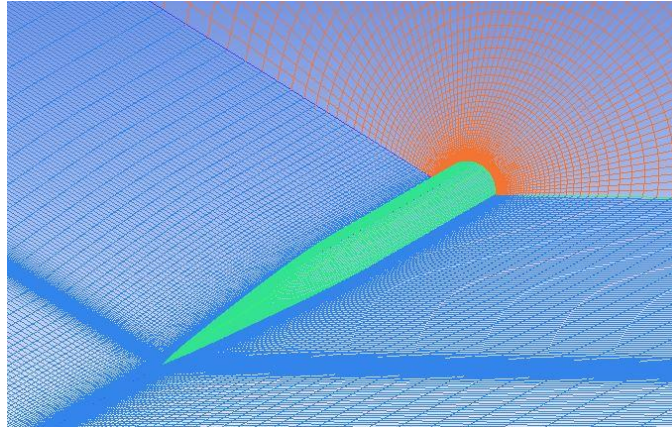
Figure 4.7: Zoomed-in view of the mesh around the randomly generated SEEB-ALR body

After the mesh is automatically generated, ICEM outputs the mesh as a .msh file to a defined path written in the replay file. The project files containing this mesh are saved. To avoid overwrites, all the mesh files and project files are deleted before the next Bezier curve is ready to be imported into the ICEM. The MATLAB closes ICEM immediately if the mesh file is found to be in the output path.

4.4 Optimization Results

4.4.1 First round of Genetic Algorithm based optimization

The first round of Genetic Algorithm optimization is completed in 19 generations. The objective function is defined as the minimization of $Max|P - Pinf|$ in the flow field. Here, $Pinf = 101325Pa$ (atmospheric pressure). Figure 4.8 shows the decrease in the fitness value ($Max|P - Pinf|$) as the generation number in SOGA increases. The fitness value gradually decreases from 817Pa to 572Pa in 19 generations. There is no change in the fitness value from 14th to 19th generation; therefore the solution is considered converged.
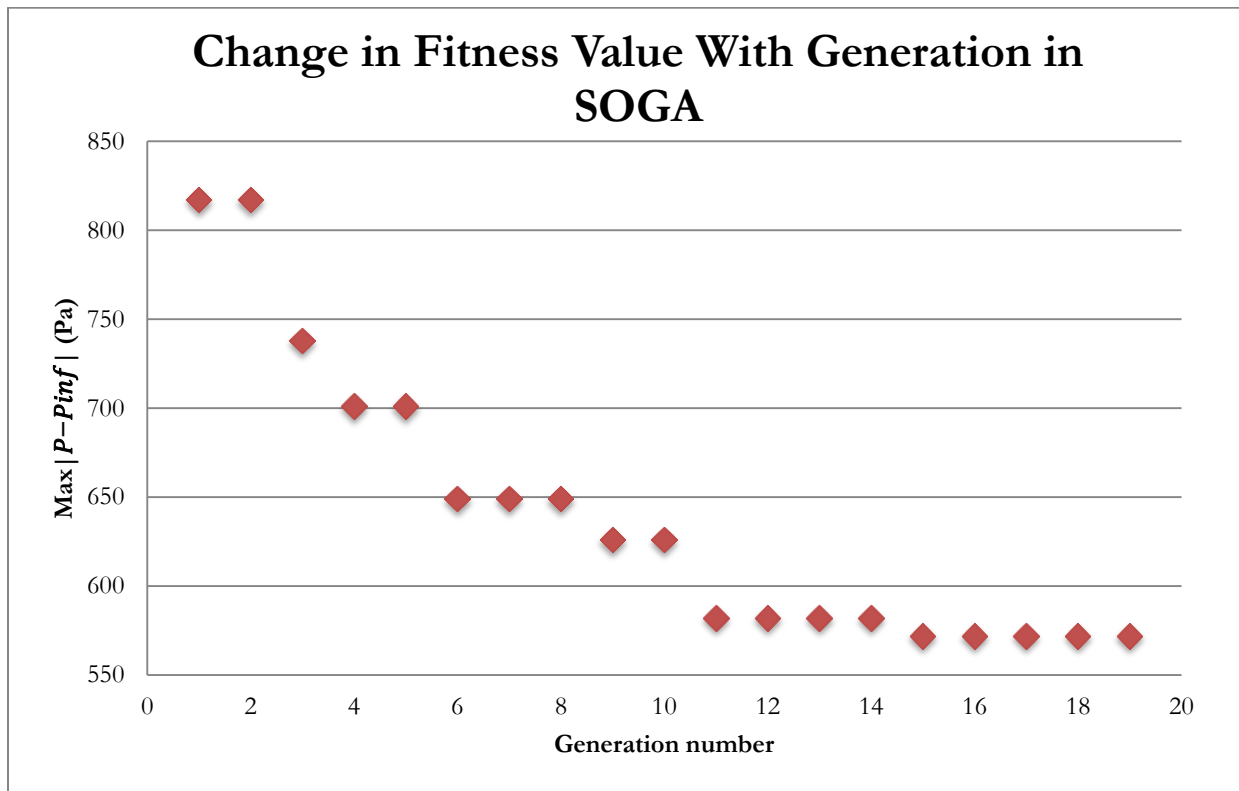
Figure 4.8: Fitness Value $|P - Pinf|$ throughout generations of 1st round in GA optimization

We consider Generation 2, Generation 5, Generation 10 and Generation 19 (the final generation in first round of GA) to see how the shape changes. Figure 4.9, Figure 4.10, Figure 4.11 and Figure 4.12 show the best shape in a given generation along with the positions of the control points.
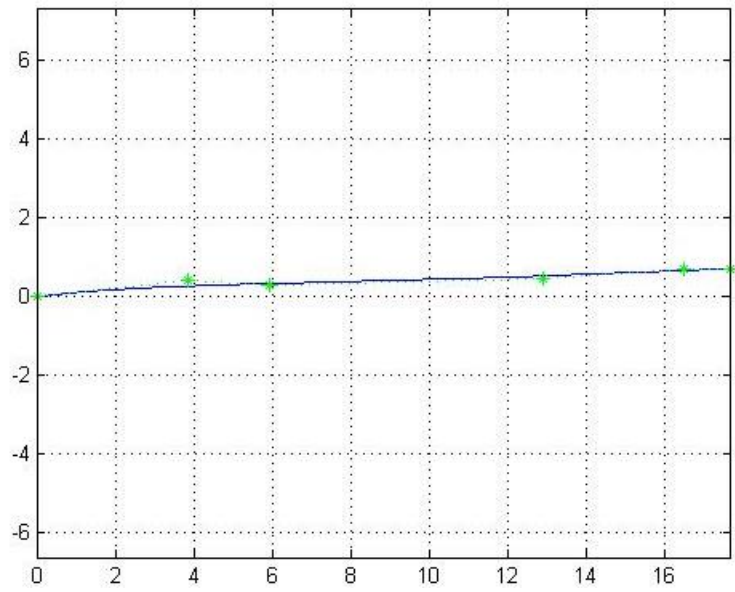
Figure 4.9: The best shape and control points in 2nd generation
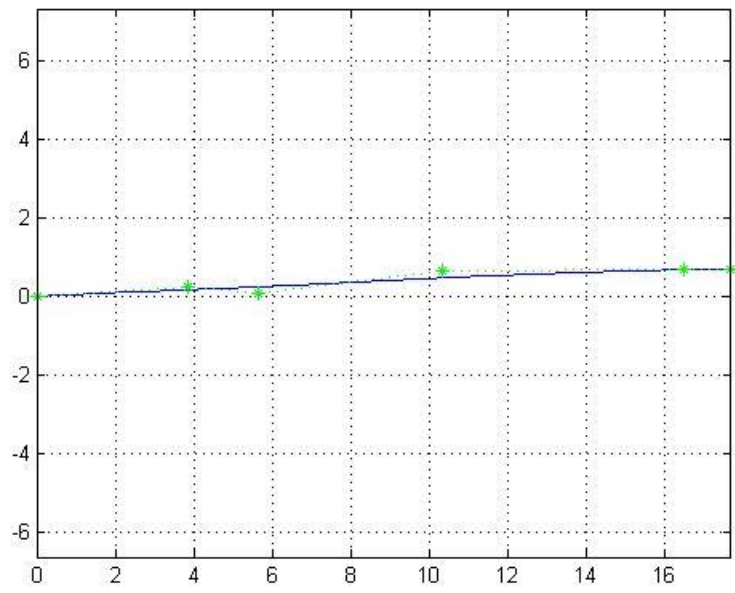
Figure 4.10: The best shape and control points in 5th generation
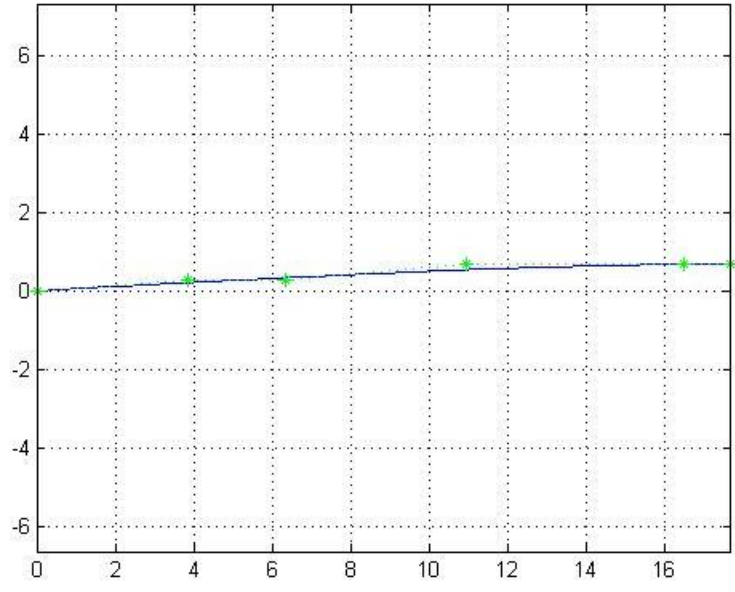
Figure 4.11: The best shape and control points in 10th generation
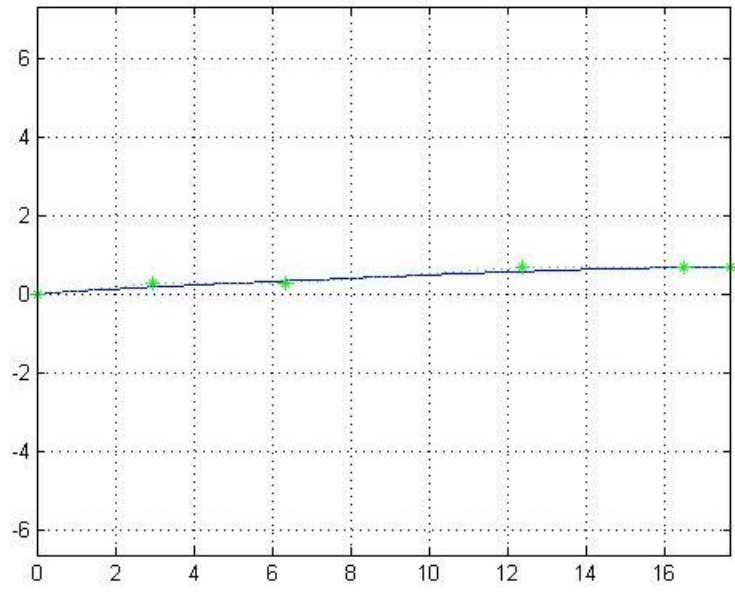
Figure 4.12: The best shape and control points in 19th generation

The control points of the best shape primarily change and move in the x-direction. The first control point gradually moves from location x=4 inches to x=3 inches, the second control point moves near x=6 inches, and the third control point moves to x=10 inches from x=13 inches and then moves back to near x=12.5 inches. In Figure 4.13, the shape of the body at different generations of SOGA

34

is shown. It should be noted that the scale in the y direction is more magnified to show the differences in shapes at various generations of SOGA. As the generation number increases, the shape of the body is becoming more and smoother. At 10th and 19th generation, the two shapes are in very small difference, which indicates that the fitness value is nearly approaching convergence after 10th generation.
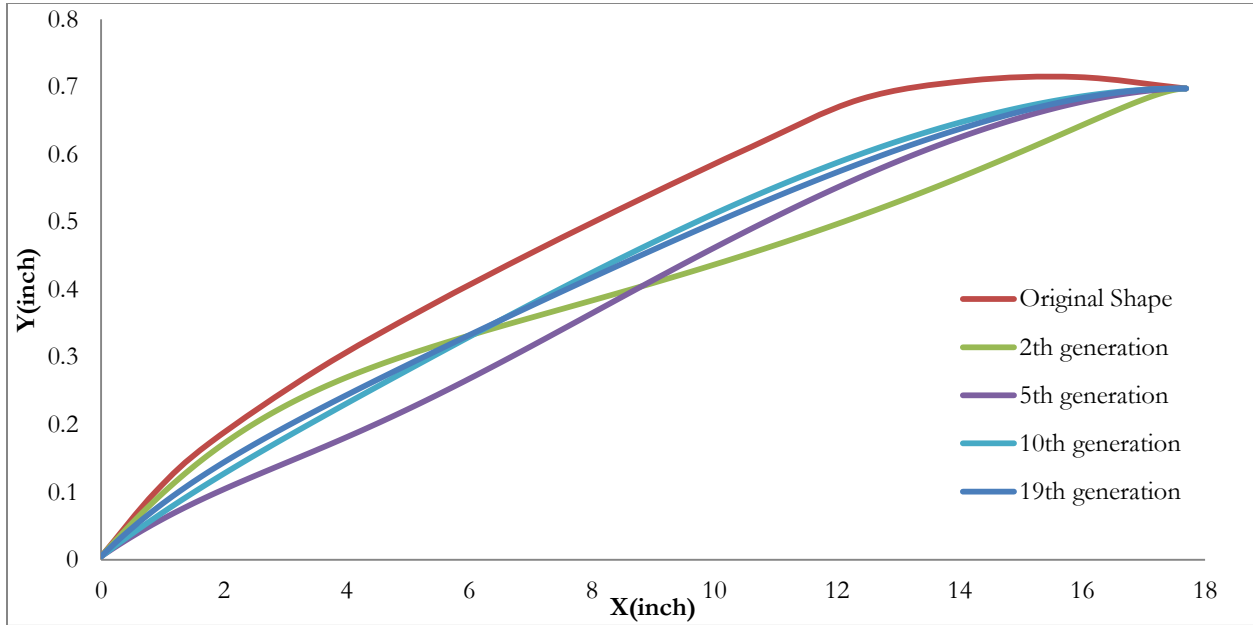
Figure 4.13: Change in body shape at various generations of SOGA

The comparison of $\Delta P/P$ for the original SEEB-ALR body and the optimized body after 19th generation of SOGA in the first round of GA are presented in Figure 4.14. The pressure disturbance of the optimized body shows a large difference compared to is that from the original body.

In case of the original body, the maximum $|P - Pinf|$ occurs at x=0.69m and the minimum occurs at x=1.1m. For the optimized body, the strength of the shock wave at these locations is significantly reduced. For the original body, the expansion wave region is from 0.93m to 1.07m; this region becomes larger from 0.85m to 1.1m for the optimized body.
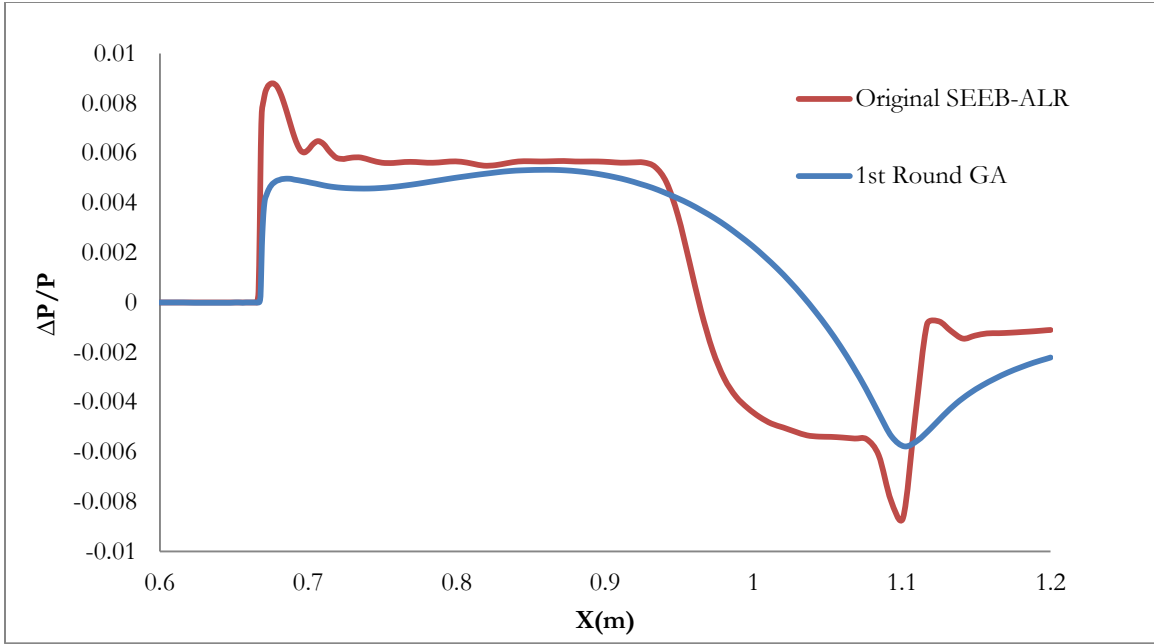
Figure 4.14: The comparison of Pressure Disturbance from original body shape along x-direction and the optimized body shape from 1ˢᵗ round of GA optimization

Table 4.3 shows the reduction in sonic boom signature using 1ˢᵗ round of Genetic Algorithm based optimization. Compared to original SEEB-ALR, the maximum value of pressure in sonic boom signature for the optimized body is reduced by 35.8%. The strength of the shock wave for the optimized body becomes weaker, resulting in noise reduction on the ground. This result can be further optimized in the next section; it is called the second round of Genetic Algorithm based optimization.

Table 4.3: Result from first round of Genetic Algorithm based optimization

	Original SEEB-ALR	Optimized SEEB-ALR
$Max\lvert P - Pinf \rvert$	892Pa	572Pa
$Max\left(\dfrac{\Delta P}{P}\right)$	0.008803	0.005645
Improvement	35.8%	

4.4.2 Second round of Genetic Algorithm based optimization

The second round of GA based optimized utilizes the optimized best shape obtained from the first round of SOGA. The window of each control point in the first round of GA was quite large, which ensured a wide range of random positions for all the control points. However the large window reduced the possibility of obtaining best optimized shape approaching the theoretically possible optimal shape. Therefore, in second round of GA based optimization, we reduce the window of each control point to be a very small region surrounding the control point positions of the optimized shape obtained from the first round.

Table 4.4: Coordinate range for each control point of the Bezier Curve for Second round of GA

Coordinate	Upper limit	Lower limit
x_1	Fixed at x=0	
x_2	2.5 inches	3.5 inches
x_3	6 inches	7 inches
x_4	10 inches	12.5 inches
x_5	Fixed at x=16.5 inches	
x_6	Fixed at x=17.692413 inches	
y_1	Fixed at y=0.005 inch	
y_2	0.25 inch	0.35 inch
y_3	0.25 inch	0.35 inch
y_4	0.6 inch	0.7 inch

y_5	Fixed at y=0.69764638
y_6	Fixed at y=0.69764638

For the second round of Genetic Algorithm based optimization, the objective is to further optimized the body shape to further minimize the peak values of pressure disturbance. The GA parameters are redefined to achieve this objective. Since the window of each control point is much smaller now, we do not need many individuals in a generation of GA. Thus the individuals in a generation are reduced, and the mutation rate is increased to have more diversity in the offspring from the parent generation. With new settings, a faster convergence is achieved towards the fitness value in this the second round of GA based optimization.

Table 4.5: Genetic Algorithm parameters for second round of GA based optimization

GA parameters	Description
Generation Size	6 individuals per generation
Number of Generations	Maximum of 50 generations if convergence is not achieved
Selection Type	Roulette Wheel Selection
Crossover Rate	0.5
Mutation Rate	0.6
Error of Mutation Constant	0.8, which determines how much mutation affects the curve as generations proceeds

The second round of Genetic Algorithm based optimization is achieved in 11 generations. The objective functions is again the minimization of $Max|P - Pinf|$ in the flow field. Here, $Pinf = 101325Pa$ (atmospheric pressure). Figure 4.15 shows the variation in fitness value $Max|P - Pinf|$ with the generation number. This time the fitness value quickly converges to 548Pa in only 5

generations. After the 5th generation, the best fitness value remains the same for 6 generations. Thus, the nearly theoretical optimized shape is definitely obtained after 11th generation of SOGA.
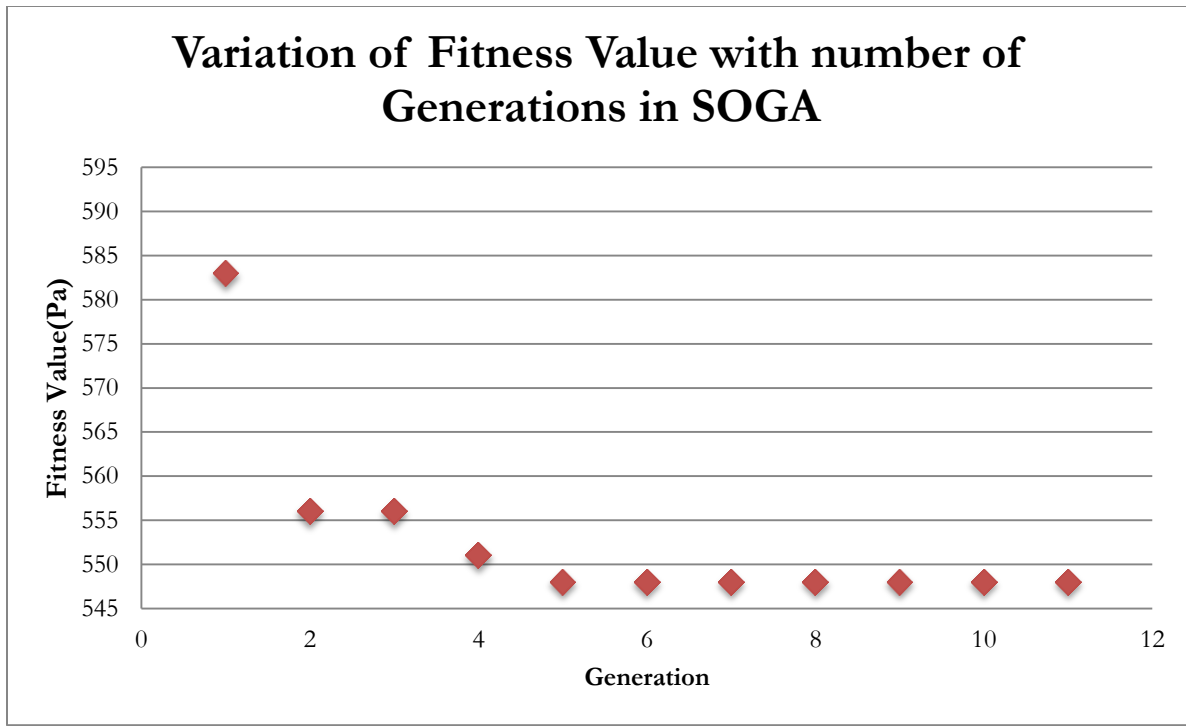
Figure 4.15: Variation of Fitness Value $|P - Pinf|$ with number of generations in 2nd round of GA based optimization

Figure 4.16 shows the body shape and its control points achieved from the 2nd round of GA based optimization. Figure 4.17 shows the comparison of the best optimized body shape of 1st and 2nd round of GA based optimization and their comparison with the original SEEB-ALR shape. Compared to the 1st round of GA based optimization, the optimized shape from the 2nd round of GA based optimization shows some more improvement.
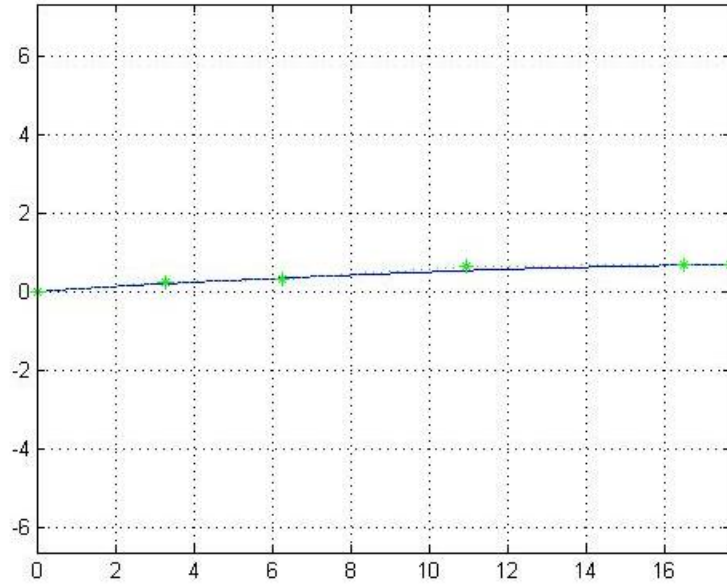
Figure 4.16: The optimal body shape and its control points from 2nd round of GA based optimization
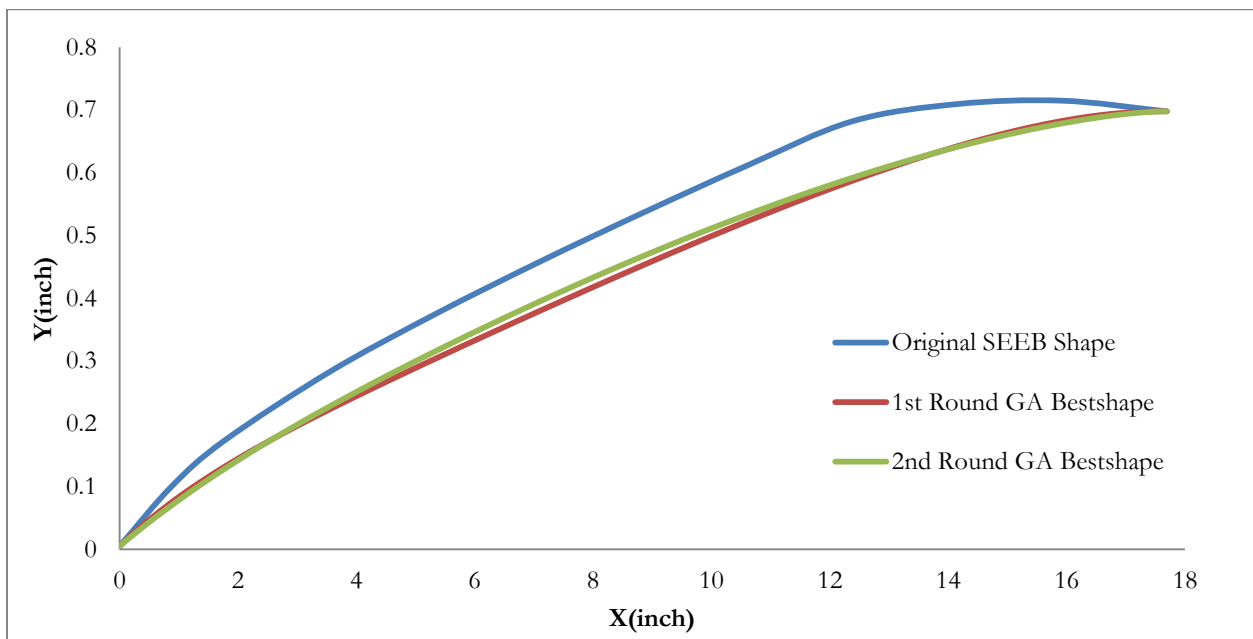
Figure 4.17: Comparison of optimized body shape from 1st round and 2nd round of GA based optimization

Figure 4.18 shows the pressure disturbance along the x-direction. The result from the 2nd round of GA based optimization is compared with the result from the 1st round of GA based optimization and with the original SEEB-ALR. Though the peak value of $|P - Pinf|$ does not show significant improvement, nevertheless the result from the 2nd round of GA based optimization shows some

improvement compared to the result from 1st round of GA based optimization. The region of expansion wave is further extended. A strong shock wave is generated which is followed by a large region of expansion wave.
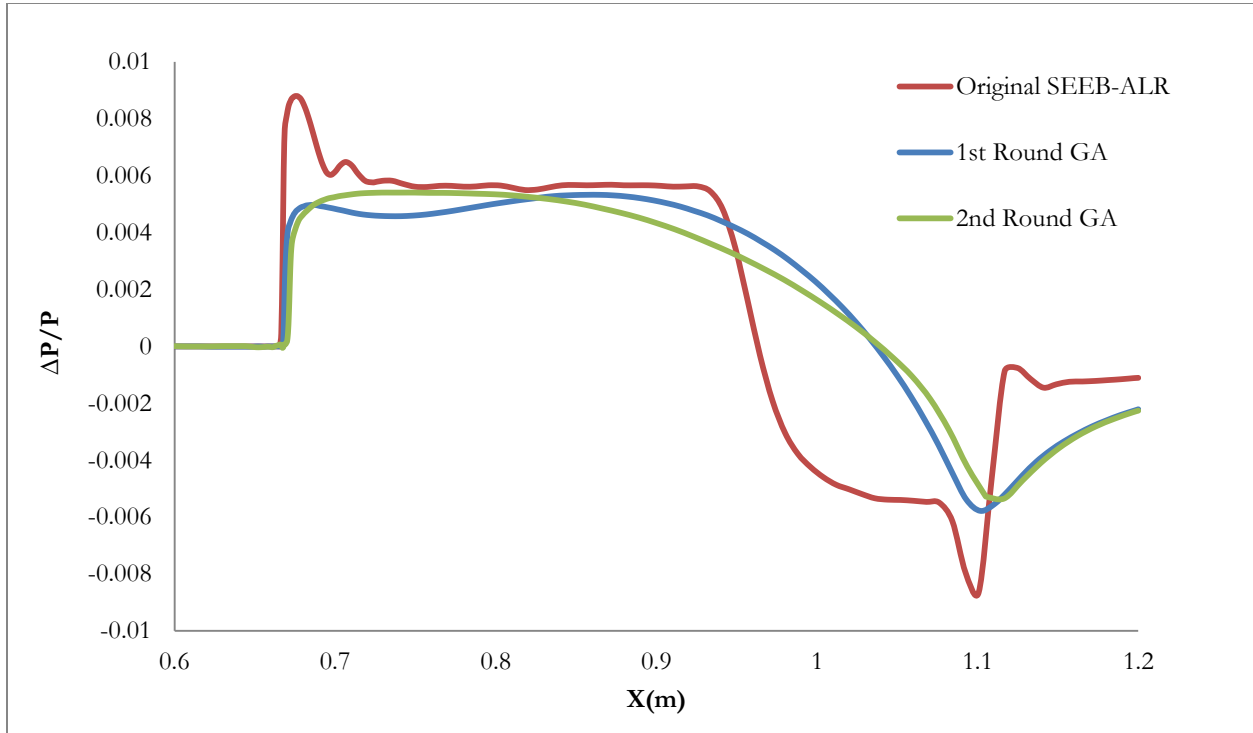
Figure 4.18: The comparison of Pressure Disturbance between optimized shape from 1st round GA optimization, 2nd round GA optimization and original shape along x-direction

Table 4.6 provides the final optimization result for both 2 rounds of Genetic Algorithm compared with the original shape. Based on the improvement reached by 1st round GA optimization, the 2nd round GA optimization further improves the fitness value by another 2.8%. As shown in Table 3.6, the net improvement in the fitness value from two rounds of GA based optimization is 38.6%; thus the sonic boom signature is significantly minimized for the final optimized SEEB-ALR body. The strength of the shock wave is reduced by nearly 40%, which is very significant in reducing the noise on the ground. The results demonstrate that CFD can be accurately and effectively employed for shape optimization of a supersonic airplane to minimize is boom signature.

Table 4.6: Optimization results from first and second round of GA based optimization

	Original SEEB-ALR	Optimized shape from 1st Round of GA	Optimized shape from 2nd Round GA
$Max\lvert P - Pinf \rvert$	892Pa	572Pa	548Pa
$Max\left(\dfrac{\Delta P}{P}\right)$	0.008803	0.005645	0.005408
Improvement	-	35.8%	38.6%

Chapter 5

Conclusion

In this thesis, numerical simulations are performed for supersonic turbulent flow past two benchmark body shapes-the Lockheed SEEB-ALR and 69° Delta Wing-Body models are researched. The geometry modeling and grid generation is conducted using ANSYS ICEM. Flow field calculations are performed using ANSYS FLUENT; compressible Reynolds-Averaged Navier-Stokes (RANS) equations in conjunction with Spalart-Allmaras (SA) model are solved. Numerical simulations are validated by comparing the results with wind-tunnel data. The computed pressure distributions agree quite well with the experimental results at all distance from the models. Shape optimization of the SEEB-ALR axisymmetric body is performed to minimize its sonic boom signature using a Single Objective Genetic Algorithm (SOGA). It is shown that a 35.8% reduction in peak pressure disturbance can be achieved by optimizing the body shape. Additional second round of optimization further reduced the peak pressure disturbance from the optimized body to 38.6% compared to that from the original SEEB-ALR body.

Chapter 6

Future work

Based on the optimization results for the SEEB-ALR model, it is established that the shape optimization is an effective way to reduce the sonic boom signature. Future works should address the more complex body shapes e.g. the 69° Delta wing-body and finally the whole supersonic airplane. Future work should also consider the propagation of sonic boom through the atmosphere to the ground using codes such as S-boom.

References

[1]   AIAA Sonic Boom Prediction Workshop

        lbpw.larc.nasa.gov/sbpw1/

[2]   Morgenstern, J., Norstrud, N., Sokhey, J., Martens, S. and Alonso, J.J., "Advanced Concept
        Studies for Supersonic Commercial Transports Entering Service in the 2018 to 2020 Period,"
        Annu. Rev. Fluid Mech, Vol. 44, 2012, pp. 505-526

[3]   Maute, K., Farhat, C., Argrow, B., Nikbay, M., "Sonic Boom Mitigation Via Shape
        Optimization Using an Adjoint Method and Application to a Supersonic Fighter Aircraft,"
        European Journal of Computational Mechanics/Revue Européenne de Mécanique
        Numérique, Vol. 17(1-2), 2008, pp. 217-243.

[4]   Chan, M. K., "Supersonic Aircraft Optimization for Minimizing Drag and Sonic Boom,"
        Diss. Stanford University, 2003, pp 2-3.

[5]   NASA   Langley   Research   Center   Turbulence   Model   Resources
        turbmodels.larc.nasa.gov/spalart.html

[6]   Goldberg, D. E., Genetic Algorithms in Search, Optimization, and Machine Learning:
        Addison-Wesley, 1989.

[7]   Morgan, B., "Gairfoils: Finding High-Lift Joukowski Airfoils with a Genetic Algorithm,"
        Technical Report, Dept. of Mechanical Engineering, Washington University in St. Louis,
        2007.

[8]   Zaman, M., and Chowdhury, S., "Modified Bezier Curves with Shape-Preserving
        Characteristics Using Differential Evolution Optimization Algorithm," Advances in
        Numerical Analysis, 2013, pp.1-3.

Brief Introduction of Appendix

Appendix A: "MOTHER.m"

This is the main program of the GA based optimization. This program controls the start and the break of optimization loops.

Appendix B: "GAparams.m"

This script controls the settings of the main parameters of GA: Number of individuals and generations, Selection type, Crossover and mutation parameters, etc.

Appendix C: "GeometryConstraints.m"

This function controls the definition of Bezier curve in the GA based optimization. Number of the control points and the constraints for each randomly generated control points is defined.

Appendix D: "Crossover.m"

This function performs the crossover operation in the GA based optimization which exchanges the information of control points between two curves.

Appendix E: "Mutator.m"

This function performs the mutation operation in the GA based optimization which allows a part of the offspring curves randomly mutate or altered.

Appendix F: "RamdomShapeLoop.m"

This script controls the generation, meshing, simulation of the random shapes before the information from simulation results entering the GA.

Appendix G: "ConvergenceCheck.m"

This function makes comparison of the results between best individuals from different generations. The convergence of result is determined by this function.

Appendix H: "ICEM Replay Control File"

This is the ICEM Replay Control File used for mesh generation. Reading this file, ICEM can automatically generate a mesh for imported Bezier curve.

Appendix I: "FLUENT Journal File"

This is the FLUNET Journal File used for flow field simulation. Reading this file, FLUNET can automatically complete the simulation and output the simulation result for GA.

Appendix A

MOTHER.m

```matlab
% by Chris Seager
% ----------------- Begin Code ----------------- %

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

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

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

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

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

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


%-------------------------------------------------------------------------
% Step 2: Randomized Shapes Loop
%-------------------------------------------------------------------------
% This is the first step for the GA: creating, simulating, and evaluating a
% generation of random individuals. Once a set of randomindividuals has
% been created, it will be saved separately from the others for future use.
% If you are restarting the GA, as noted in the prompt for "initial run,"
% this entire step will be skipped and the code will begin at step 4.

if strcmp(initRun,'Y') == 1 || (strcmp(initRun,'Y') == 0 && generation == 1
&& numTrial <= genSize)
   RandomShapesLoop
end
```

```
%-------------------------------------------------------------------------
% Step 3: Genetic algorithm
%-------------------------------------------------------------------------
% Now that we have our generation of random curves and their data stored in
our
% global matrices, we can run the genetic algorithm (GA) and then repeat for
% a loop using daughter curves. The genetic algorithm is simple once you
% understand it, but calls multiple functions. Read carefully.

if strcmp(initRun,'Y') == 1 || (strcmp(initRun,'Y') == 0 && generation == 1
&& numTrial == genSize+1)
    MOGAgeneral
    ResetVars
end


%-------------------------------------------------------------------------
% Step 4: Loop until converged
%-------------------------------------------------------------------------
% After the above 3 steps we now have our cell array "BEZIER_CURVES"
% containing all of the CP information for the newly generated curves. We
% also have saved the best curve's information. Now we cycle through a very
% similar loop to "RandomShapeLoop", except our "RandomIndividuals"
% function is replaced with something that is no longer random. Everything
% else is, for the most part, the same.

while  ConvergenceCheck(bestCurveVals,nDV,maxGen,minGen)  ==  0  %  while  the
solution is not converged
    OptimizationLoop        % run the in-generation loop
    MOGAgeneral
    ResetVars
end

totalRuns = generation*genSize %#ok<NOPTS>


%-------------------------------------------------------------------------
% end of script
```

Appendix B

GAparams.m

```matlab
% GAparams
% GAparams is a script to take in your parameters for the GA. The variables
% in here need to be individualized for your case.

% -----------------------------------------------------------------------------
% Step 1: Individuals per generation and maximum generations
% -----------------------------------------------------------------------------
% 1a) Define generation size (number of individuals per generation)
genSize = 6; %ind    % must be greater than 1 and an even integer.

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

% 1b) Define maximum generations for the optimization
maxGen = 20; %ind    % must be greater than 1 and an integer

% 1c) Define minimum generations for optimization
minGen = 19; %ind    % must be greater than 1 and an integer

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

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

% -----------------------------------------------------------------------------
% Step 2: Number of design variables (dVs)
% -----------------------------------------------------------------------------
% 2a) Define the number of design variables
% Number of design variables (nDV) is how many variables you need to keep
% track of to use in your fitness function. Normally, these are every
% variable that is being output by your FLUENT file.

nDV = 1; %ind

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

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

dvObj = cell(1,nDV); % allocates space

% Input by using the following pattern. dvObj{#} = 'min' or 'max'
dvObj{1} = 'min'; %ind
%dvObj{3} = 'max'; %ind

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


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

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

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

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

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


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

```matlab
crossRate = .5; %ind

% makes sure the rate is an acceptable number
if crossRate >= 1 || crossRate <= 0
    error('Crossover rate is outside allowable bounds')
end

% 4b) Mutation rate
% This is the probability that any given variable control point is mutated.
% It helps increase genetic diversity to avoid local optimums. It must be
% between 0 and 1. When you have many variable control points, mutRate
% should be closer to 0, when you have few variable control points, it can
% increase towards .5 or higher.

mutRate = .6; %ind

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

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

eomC = .8; %ind

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

%-------------------------------------------------------------------------
% end of script
```

Appendix C

GeometryConstraints.m

```matlab
function                [Mcurves,Ncurves,nBC,fCPinfo,vCPinfo,limCPinfo]                =
GeometryConstraints
% GeometryConstraints is a key function for your GA. This is where you
% define your number of Bezier curves (BCs), the amount of control points
% (CPs), the fixed points, and boundaries for the possible locations of
% the variable control points.

% These boundaries will affect the range of possible curves that can be
% generated or mutated, which has implications for the mesh, simulation,
% and final results: too small of boundaries, and although your mesh and
% simulation will likely give good results, your solution may be a local
% optimum. Too large of boundaries, and while you are encompassing more
% possible solutions, your mesh and simulation might cause errors. THESE
% BOUNDARIES ARE DEFINED BASED ON YOUR JUDGEMENT. MAKE THEM AS LARGE AS
% POSSIBLE TO ENCOMPASS THE MOST CURVE POSSIBILITES, BUT NOT SO LARGE THAT
% YOU ENCOMPASS PRACTICALLY IMPOSSIBLE CURVES OR CAUSE MESHING/SIMULATION
% ERROR.


%-------------------------------------------------------------------------
% Step 1: Identify Bezier curve sizes and location and value of fixed CPs
%-------------------------------------------------------------------------
% Input each of your Bezier curves as 2 x nCP (or 3 x nCP if you're doing
% 3 dimensional and then by god good luck) where nCP is the total number of
% control points. Use "nan" for a variable point and the numerical value for
% a fixed point. The first row is "x" values and the second row is "y"
% values. You can do this for as many or as few Bezier curves as you like,
% as long as you maintain the format and variable names I have below.
%

% Number of Bezier curves you are using
nBC = 1; %ind

% BP is a cell array for your BCs. It scales to whichever size you need.
BP = cell(1,nBC);

% Define your curves here, one matrix for each cell in BP
BP{1}                                                                    =
[0,nan,nan,nan,16.5,17.692413;.005,nan,nan,nan,0.69764638,0.69764638]; %ind
%BP{2} = [0,nan,nan,nan,.04;.002,nan,nan,.006,.006];

% If you have more than one BC, the format is BP{#} = [x values; y values];
% I have some example curves for this below.

% BP{2} = [-10,nan,-5,0;-5,nan,nan,0]; %ind
% BP{3} = [0,nan,nan,nan,5;0,nan,nan,nan,10]; %ind
```

```matlab
% Checks to make sure you have the right number of curves defined
if sum(cellfun('isempty',BP)) ~= 0
    error('Defined Bezier curves do not match defined number of Bezier
curves')
end

% Defining necessary parameters from your curves (does so automatically)
Ncurves = zeros(1,nBC);      % holds the number of rows for each curve
Mcurves = Ncurves;           % holds the number of cols (nCP) for each curve
vCPinfo = cell(1,nBC);        % holds the index of variable control points
(vCPs) for each curve
fCPinfo = cell(2,nBC);       % holds the index and values of fixed control
points (fCPs) for each curve
for i = 1:nBC    % puts in the values for all of the above variables
    [Ncurves(i) Mcurves(i)] = size(BP{i});
    vCPinfo{i} = find(isnan(BP{i})==1)';
    fCPinfo{1,i} = find(isnan(BP{i})==0)';
    fCPinfo{2,i} = BP{i}(fCPinfo{1,i});
end


%-------------------------------------------------------------------------
% Step 2: Define the "box" limits for each variable points
%-------------------------------------------------------------------------
% This step is also based on your geometry. Input your values for each "box
% limit" for each variable control point. The array "vCPinfo" contains the
% matrix index of your variable values. Make sure you are familiar with
% MATLAB matrix indexing: 1 is the first row and column (x1), 2 is the
% second row and first column (y1), 3 is the first row and second column (x2),
% 4 is the second row and second column (y2)... and so on. To make it easy
% to number, output vCPinfo{x} (x = 1,2,...,nBC) and see which indices are
variable.
% Then assign those indices limits. It helps to physically draw out your
% limits first and then input them here.

% Curve 1
%                                  BP{1}                                  =
[0,nan,nan,nan,16.5,17.692413;.005,nan,nan,nan,0.69764638,0.69764638]; %ind
limsC1(1,:) = [2.5,3.5];
limsC1(2,:) = [0.25,0.35];
limsC1(3,:) = [6,7];
limsC1(4,:) = [0.25,0.35];
limsC1(5,:) = [10,12.5];
limsC1(6,:) = [0.6,0.7];

% % Curve 2
% % BP{2} = [0,nan,nan,nan,.04;.002,nan,nan,.006,.006];
% limsC2(1,:) = [0,.013];
% limsC2(2,:) = [.002,.012];
% limsC2(3,:) = [.013,.026];
% limsC2(4,:) = [.002,.012];
% limsC2(5,:) = [.026,.04];
```

```matlab
% If you have more than one curve, follow the formatting given below.
% % Curve 2
% limsC2(1,:) = [-10,-5]; %ind    %this is the limit for curve 2 index 3, or
x2 of BP1
% limsC2(2,:) = [-5,0];   %ind    %this is the limit for curve 2 index 4, or
y2 of BP1
% limsC2(3,:) = [-10,5];  %ind    %this is the limit for curve 2 index 6, or
y3 of BP1


%
% % Curve 3
% limsC3(1,:) = [0,5/3];     %ind %this is the limit for curve 3 point 3, or
x2 of BP2
% limsC3(2,:) = [-10,10];    %ind %this is the limit for curve 3 point 4, or
y2 of BP2
% limsC3(3,:) = [5/3,10/3];  %ind %this is the limit for curve 3 point 5, or
x3 of BP2
% limsC3(4,:) = [-10,10];    %ind %this is the limit for curve 3 point 6, or
y2 of BP2
% limsC3(5,:) = [10/3,5];    %ind %this is the limit for curve 3 point 7, or
x4 of BP2
% limsC3(6,:) = [-10,10];    %ind %this is the limit for curve 3 point 8, or
y4 of BP2

% Makes sure our curve limits are defined correctly
for i = 1:nBC
    if length(eval(sprintf('limsC%d',i))) ~= length(vCPinfo{i})
        error('Defined limits do not agree with number of variable control
points from defined Bezier curves')
    end
end

% Storing the imits for each variable CP for each curve
limCPinfo = cell(1,nBC);
for i = 1:nBC
    limCPinfo{i} = eval(sprintf('limsC%d',i));
end


end

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

Appendix D

Crossover.m

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

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

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

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


end

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

Appendix E

Mutator.m

```
function                            [DAUGHTER_CURVES]                        =
Mutator(vCPinfo,limCPinfo,mutRate,genSize,DAUGHTER_CURVES,generation,maxGen,e
omC)
% Mutator takes in the new daughter curves and iterates over all of their
% variable CPs. mutRate % of those vCPs will be mutated, or randomly
% altered.

% -------------------------------------------------------------------------
% Step 1: Define the effective mutation efficiency (eom)
% -------------------------------------------------------------------------
% We want eom to start at 1 and decrease to eomC by maxGen

eom = 1 - eomC*generation/maxGen;

% -------------------------------------------------------------------------
% Step 2: Mutate the curves
% -------------------------------------------------------------------------
nBC = length(DAUGHTER_CURVES);

for i = 1:genSize % iterate over all individuals
    for j = 1:nBC % iterate over each bezier curve
        curve2Mut = DAUGHTER_CURVES{j}(2*i-1:2*i,:); % define curve CPs
        for k = 1:length(vCPinfo{j}) % iterate over all vCPs
            doMutation = (rand(1) <= mutRate);  % mutation probability is
mutRate
            if doMutation == 1
                val2Mut = curve2Mut(vCPinfo{j}(k)); % value of point being
mutated
                if rand(1)<.5
                    edgeVal = limCPinfo{j}(k,1);    % select one edge value
                else
                    edgeVal = limCPinfo{j}(k,2);    % otherwise select other
                end
                curve2Mut(vCPinfo{j}(k))  =  val2Mut  +  eom*rand(1)*(edgeVal-
val2Mut); %curve moves some random distance towards that edge
            end
        end
        DAUGHTER_CURVES{j}(2*i-1:2*i,:) = curve2Mut; % replacing in DC matrix
    end
end

end

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

Appendix F

RandomShapeLoop.m

```matlab
% RandomShapesLoop
% This script contains the first generation of the GA, the randomized
% shapes. genSize random shapes will be generated, meshed, and simulated
% before entering the GA.

% -------------------------------------------------------------------------
% Step 1: Print parameters
% -------------------------------------------------------------------------
fprintf('Generation %i\n',generation)    %prints  generation  number  to  command
window

%prints generation number to "Results.txt"
fid = fopen('Results.txt','w');
fprintf(fid,'Generation %i\n',generation);
fclose(fid);

% -------------------------------------------------------------------------
% Step 2: Generating, meshing, and simulating individuals
% -------------------------------------------------------------------------
while numTrial <= genSize    %sets up a loop to run through specified number
of trials.
    fprintf('Trial: %i\n',numTrial)          %prints  trial  number  to  command
window

    % RandomIndividuals generates your point data to be loaded into ICEM
    BEZIER_POINTS                                                          =
RandomIndividuals(numTrial,Ncurves,Mcurves,fCPinfo,vCPinfo,limCPinfo,BEZIER_P
OINTS);

    % Note: After working with Felix and changing RandomIndividuals for a
    % complicated geometry, I believe it would be much easier to generate
    % any extra geometries here instead of inside that function. So, if you
    % need to generate geometries based off of your Bezier curves do so
    % here. All of your BC info is saved in files called "Points_#" The
    % following lines (if uncommented) will extract that point data.

    % points = cell(1,nBC);     % allocate space
    % for i = 1:nBC             % over all Points_# 's
    %    filename = sprintf('%s_%d','Points',i); % define the filename
    %    points{i} = dlmread(filename);          % load in the data
    %    points{i} = points{i}(2:end,1:3);       % omit ICEM headlines
    % end

    % Now, "points" is a cell array, and in each cell is a matrix with "x"
    % values of your BCs in the first column, and the "y" values in the
```

```
    % second column. Any extra points files you need to create can now be
    % done by using these files here. For instance,

    % PointsMirror1 = points{1};
    % PointsMirror1(:,2) = (points{1}(:,2))*-1
    % fid = fopen('PointsMirror_1','w');          % opens or creates the file
and writes info.
    % fprintf(fid,'%5.0f %1.0f\n', length(PointsMirror1), 1); % Format ICEM
needs to load
    % fprintf(fid,'\t\t%f\t%f\t%f\n', points);         % Format ICEM needs to
load
    % fclose(fid);    %close the file id

    % Those 6 lines would flip the "y" values of the first BC and save
    % the new file.

    pause(2); %give program time to catch up

    delete SEEB-ALR.msh       %this deletes the file MATLAB will wait for from
ICEM.

    % IcemReplay calls your icem script to mesh the individual
    IcemReplay

    % MeshWait waits for you mesh to be created. If it isn't, this case
    % will be thrown out.
    haveMesh = MeshWait(maxMeshTime);

    % Now we should have a mesh. FLUENT can load the mesh, so we need to call
    % FLUENT and the journal file for the program. This is done in the
    % function "FluentSimulator".

    if             haveMesh              ==              1              &&
FluentSimulator(runCheckTime,maxSimTime,updateTime,nDV) == 1; % makes sure we
have the mesh (haveMesh == 1) and the simulation ran correctly
    % if FLUENT ran correctly, we will load in our data.

    % %importfile('file') will read your FLUENT output and separate the
    % text into a variable called "textdata" and your numbers into a
    % variable called "data."

    % the "try" and "catch" statements below are built in for redundancy.
    % Basically, unless everything goes right (the meshing, simulation, and
    % reading of data) NullValues will be called.

% -------------------------------------------------------------------------
% Step 3: Loading in design variables
% -------------------------------------------------------------------------
% We've generated shapes, meshed them, and simulated them according to our
% journal file. FLUENT output files with your data of interest, and now we
% need to read those files in for the GA. The following loop does so.
% "importfile" is a MATLAB function that separates text and numerical data
% from your DV files, and you don't need to adjust it. "importDVs" however,
% is a file you need to individualize.
```

```matlab
        try %sometimes there is an error in loading. If so, we scrap the case.
            dvStore = cell(1,nDV);
            for i = 1:nDV % iterate over every design variable
                filename = sprintf('%s_%d','FLUENT_DES_VAR',i);
                importfile(filename)
                dvStore{i} = data; %now dvStore has each dv's data of
interest
            end
            DES_VARS = ImportDVs(numTrial,dvStore,DES_VARS);
        catch
            fprintf('calling NullValues\n');
            [DES_VARS] = NullValues(numTrial,dvObj,DES_VARS,OLD_DES_VARS);
        end
    else % if the mesh or simulation failed, save the design variables as bad
values.
        [DES_VARS] = NullValues(numTrial,dvObj,DES_VARS,OLD_DES_VARS);
        fprintf('calling NullValues in second stage\n');
    end

    pause(5); %gives time for any functions to catch up.

    numTrial = numTrial + 1;    %reiterate

end

% Saving data before the GA.
SaveCurves(BEZIER_POINTS,DAUGHTER_CURVES)
OLD_DES_VARS = DES_VARS; % These will be the same as this is the initial
generation
SaveDVs(DES_VARS,OLD_DES_VARS)

%-------------------------------------------------------------------------
% end of script
```

Appendix G

CovergenceCheck.m

```matlab
function [converged] = ConvergenceCheck(bestCurveVals,nDV,maxGen,minGen)
% ConvergenceCheck is a function that reads in the best individuals
% information and compares it to the best individuals from all previous
% generations. By comparing the resulting shapes and their DVs we can
% easily visualize convergence.

%bestCurveVals = dlmread('ConvergenceData.txt'); % in case of needing backup,
can load from text file
 %for i = 1:nBC
  %    filename = sprintf('%s_%d','bestCurve',i);
  %    bestCurve{i} = dlmread(filename);
%end


% ------------------------------------------------------------------------
% Step 1: Plot the data
% ------------------------------------------------------------------------
% % best possible was for my initial test
% bestPossible = zeros(generation,2);
% bestPossible(:,1) = 1:generation;
% bestPossible(:,2) = 32.36068;



[generation,~] = size(bestCurveVals);

for i = 1:nDV
    subplot(1,nDV,i)
    scatter(1:generation,bestCurveVals(:,i))
    titString = sprintf('Design Variable %d Throughout Generations',i);
    title(titString);
    xlabel('Generation');
    ylabel(['Design Variable ',i])
    set(gca,'xtick',0:5:generation);
%      hold on
%      plot(bestPossible(:,1),bestPossible(:,2))

end


% ------------------------------------------------------------------------
% Step 2: Determine Convergence
% ------------------------------------------------------------------------

bestVal = bestCurveVals(end);
if generation > minGen
    bestAvgVal = sum(bestCurveVals(end-4:end))/(4+1);
else
```

```matlab
    bestAvgVal = inf;
end

if generation >= maxGen || strcmp(num2str(bestVal,3),num2str(bestAvgVal,3))
== 1 %bestVal < 32.4 && bestAvgVal < 32.5
    converged = 1;
else
    converged = 0;
end

end

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

Appendix H

ICEM Replay Control File

```
ic_undo_group_begin
ic_geo_cre_geom_input F:/GAcode/Points 0.000001 input PNTS pnt CRVS crv SURFS
{}
ic_boco_solver
ic_boco_clear_icons
ic_csystem_display all 0
ic_csystem_set_current global
ic_boco_nastran_csystem reset
ic_undo_group_end
ic_undo_group_begin
ic_geo_new_family GEOM
ic_boco_set_part_color GEOM
ic_point {} GEOM pnt.00 0,0,0
ic_undo_group_end
ic_undo_group_begin
ic_point {} GEOM pnt.01 0,-0.005,0
ic_undo_group_end
ic_undo_group_begin
ic_point {} GEOM pnt.02 29.264446,0.69736552,0
ic_undo_group_end
ic_undo_group_begin
ic_point {} GEOM pnt.03 29.264446,-0.69736552,0
ic_undo_group_end
ic_undo_group_begin
ic_point {} GEOM pnt.04 29.264446,0,0.69736552
ic_undo_group_end
ic_undo_group_begin
ic_undo_group_end
ic_undo_group_begin
ic_delete_geometry curve names crv.00 0
ic_curve point PNTS crv.00 {pnt99 pnt.02}
ic_undo_group_end
ic_undo_group_begin
ic_undo_group_end
ic_undo_group_begin
ic_point {} GEOM pnt.05 29.264446,0,0
ic_undo_group_end
ic_undo_group_begin
ic_undo_group_end
ic_undo_group_begin
ic_geo_cre_srf_rev CRVS srf.00 {crv0 crv.00} pnt.00 {1 0 0} 0 -180 c 1
ic_set_dormant_pickable point 0 {}
ic_set_dormant_pickable curve 0 {}
ic_undo_group_end
ic_undo_group_begin
ic_undo_group_end
ic_undo_group_begin
```

```
ic_point {} GEOM pnt.06 0,0,0.005
ic_undo_group_end
ic_undo_group_begin
ic_undo_group_end
ic_undo_group_begin
ic_curve arc PNTS crv.01 {pnt0 pnt.06 pnt.01}
ic_undo_group_end
ic_undo_group_begin
ic_undo_group_end
ic_undo_group_begin
ic_undo_group_end
ic_undo_group_begin
ic_undo_group_end
ic_undo_group_begin
ic_undo_group_end
ic_undo_group_begin
ic_delete_geometry curve names crv.02 0
ic_curve point PNTS crv.02 {pnt0 pnt.01}
ic_undo_group_end
ic_undo_group_begin
ic_undo_group_end
ic_undo_group_begin
ic_surface 2-4crvs PNTS srf.01 {0.01 {crv.01 crv.02}}
ic_set_dormant_pickable point 0 {}
ic_set_dormant_pickable curve 0 {}
ic_undo_group_end
ic_undo_group_begin
ic_point {} GEOM pnt.07 -15,0,0.005
ic_undo_group_end
ic_undo_group_begin
ic_point {} GEOM pnt.08 -15,0.005,0
ic_undo_group_end
ic_undo_group_begin
ic_point {} GEOM pnt.09 -15,-0.005,0
ic_undo_group_end
ic_undo_group_begin
ic_undo_group_end
ic_undo_group_begin
ic_curve arc GEOM crv.03 {pnt.08 pnt.07 pnt.09}
ic_undo_group_end
ic_undo_group_begin
ic_undo_group_end
ic_undo_group_begin
ic_undo_group_end
ic_undo_group_begin
ic_delete_geometry curve names crv.04 0
ic_curve point GEOM crv.04 {pnt.08 pnt.09}
ic_undo_group_end
ic_undo_group_begin
ic_undo_group_end
ic_undo_group_begin
ic_surface 2-4crvs GEOM srf.02 {0.01 {crv.03 crv.04}}
ic_set_dormant_pickable point 0 {}
ic_set_dormant_pickable curve 0 {}
ic_undo_group_end
ic_undo_group_begin
ic_undo_group_end
ic_undo_group_begin
ic_delete_geometry curve names crv.05 0
```

```
ic_curve point GEOM crv.05 {GEOM.9 PNTS.6}
ic_undo_group_end
ic_undo_group_begin
ic_undo_group_end
ic_undo_group_begin
ic_delete_geometry curve names crv.06 0
ic_curve point GEOM crv.06 {pnt.09 pnt.01}
ic_undo_group_end
ic_undo_group_begin
ic_undo_group_end
ic_undo_group_begin
ic_point {} GEOM pnt.10 109.4,-100,0
ic_undo_group_end
ic_undo_group_begin
ic_point {} GEOM pnt.11 109.4,100,0
ic_undo_group_end
ic_undo_group_begin
ic_point {} GEOM pnt.12 154.08446,100,0
ic_undo_group_end
ic_undo_group_begin
ic_point {} GEOM pnt.13 154.08446,-100,0
ic_undo_group_end
ic_undo_group_begin
ic_point {} GEOM pnt.14 154.08446,0,100
ic_undo_group_end
ic_undo_group_begin
ic_point {} GEOM pnt.15 109.4,0,100
ic_undo_group_end
ic_undo_group_begin
ic_undo_group_end
ic_undo_group_begin
ic_curve arc GEOM crv.07 {pnt.11 pnt.15 pnt.10}
ic_undo_group_end
ic_undo_group_begin
ic_undo_group_end
ic_undo_group_begin
ic_curve arc GEOM crv.08 {pnt.12 pnt.14 pnt.13}
ic_undo_group_end
ic_undo_group_begin
ic_undo_group_end
ic_undo_group_begin
ic_undo_group_end
ic_undo_group_begin
ic_delete_geometry curve names crv.09 0
ic_curve point GEOM crv.09 {pnt.09 pnt.10}
ic_undo_group_end
ic_undo_group_begin
ic_undo_group_end
ic_undo_group_begin
ic_delete_geometry curve names crv.10 0
ic_curve point CRVS crv.10 {CRVS.13 pnt.13}
ic_undo_group_end
ic_undo_group_begin
ic_undo_group_end
ic_undo_group_begin
ic_delete_geometry curve names crv.11 0
ic_curve point GEOM crv.11 {pnt.02 pnt.12}
```

```
ic_undo_group_end
ic_undo_group_begin
ic_undo_group_end
ic_undo_group_begin
ic_delete_geometry curve names crv.12 0
ic_curve point GEOM crv.12 {GEOM.9 pnt.11}
ic_undo_group_end
ic_undo_group_begin
ic_undo_group_end
ic_undo_group_begin
ic_delete_geometry curve names crv.13 0
ic_curve point GEOM crv.13 {pnt.10 pnt.13}
ic_undo_group_end
ic_undo_group_begin
ic_undo_group_end
ic_undo_group_begin
ic_delete_geometry curve names crv.14 0
ic_curve point GEOM crv.14 {pnt.11 pnt.12}
ic_undo_group_end
ic_undo_group_begin
ic_undo_group_end
ic_undo_group_begin
ic_surface  2-4crvs  CRVS  srf.03  {0.01  {crv0  crv.00  crv.11  crv.14  crv.12
crv.05}}
ic_set_dormant_pickable point 0 {}
ic_set_dormant_pickable curve 0 {}
ic_undo_group_end
ic_undo_group_begin
ic_surface  2-4crvs  CRVS  srf.04  {0.01  {srf.00e4  srf.00.1e7  crv.10  crv.13
crv.09 crv.06}}
ic_set_dormant_pickable point 0 {}
ic_set_dormant_pickable curve 0 {}
ic_undo_group_end
ic_undo_group_begin
ic_surface 2-4crvs GEOM srf.05 {0.01 {crv.06 crv.02 crv.05 crv.04}}
ic_set_dormant_pickable point 0 {}
ic_set_dormant_pickable curve 0 {}
ic_undo_group_end
ic_undo_group_begin
ic_geo_cre_srf_loft_crvs CRVS srf.06 0.001 {srf.00.1e6 crv.08} 4 0 1
ic_set_dormant_pickable point 0 {}
ic_set_dormant_pickable curve 0 {}
ic_undo_group_end
ic_undo_group_begin
ic_undo_group_end
ic_undo_group_begin
ic_curve arc GEOM crv.01 {GEOM.17 pnt.07 GEOM.18}
ic_undo_group_end
ic_undo_group_begin
ic_undo_group_end
ic_undo_group_begin
ic_geo_cre_srf_loft_crvs GEOM srf.07 0.001 {crv.01 crv.07} 4 0 1
ic_set_dormant_pickable point 0 {}
ic_set_dormant_pickable curve 0 {}
ic_undo_group_end
ic_undo_group_begin
ic_surface 2-4crvs GEOM srf.08 {0.01 {crv.08 crv.14 crv.07 crv.13}}
```

```
ic_set_dormant_pickable point 0 {}
ic_set_dormant_pickable curve 0 {}
ic_undo_group_end
ic_undo_group_begin
ic_undo_group_begin
ic_geo_duplicate_set_fam_and_data curve srf.00e4 srf.00e4.0 {} _0
ic_geo_duplicate_set_fam_and_data curve srf.00.1e7 srf.00.1e7.0 {} _0
ic_move_geometry curve names {srf.00e4.0 srf.00.1e7.0} rotate -60 rotate_axis
{1 0 0} cent {0 0 0}
ic_undo_group_end
ic_geo_duplicate_set_fam_and_data curve srf.00e4 srf.00e4.1 {} _1
ic_geo_duplicate_set_fam_and_data curve srf.00.1e7 srf.00.1e7.1 {} _1
ic_move_geometry   curve   names   {srf.00e4.1   srf.00.1e7.1}   rotate   -120
rotate_axis {1 0 0} cent {0 0 0}
ic_undo_group_end
ic_geo_reset_data_structures
ic_undo_group_end
ic_geo_configure_one_attribute surface shade wire
ic_undo_group_begin
ic_geo_new_family FLUID
ic_boco_set_part_color FLUID
ic_hex_initialize_blocking {} FLUID 0 101
ic_hex_switch_blocking root
ic_hex_unblank_blocks
ic_hex_multi_grid_level 0
ic_hex_projection_limit 0
ic_hex_default_bunching_law default 2.0
ic_hex_floating_grid off
ic_hex_transfinite_degree 1
ic_hex_unstruct_face_type one_tri
ic_hex_set_unstruct_face_method uniform_quad
ic_hex_set_n_tetra_smoothing_steps 20
ic_hex_set_mesh_params PNTS CRVS GEOM TEMP_TSF_FAM FLUID -version 110
ic_hex_error_messages off_minor
ic_hex_switch_blocking root
ic_undo_group_end
ic_hex_mark_blocks unmark
ic_undo_group_begin
ic_hex_mark_blocks superblock 13
ic_undo_group_end
ic_undo_group_begin
ic_hex_mark_blocks face_neighbors corners { 21 25 22 26 } { 37 41 38 42 }
{ 21 37 25 41 }
ic_undo_group_end
ic_undo_group_begin
ic_hex_ogrid 1 m PNTS CRVS GEOM TEMP_TSF_FAM FLUID -version 50
ic_hex_mark_blocks unmark
ic_undo_group_end
ic_hex_mark_blocks unmark
ic_undo_group_begin
ic_hex_set_edge_projection 74 75 0 1 crv.01
ic_hex_set_edge_projection 72 75 0 1 crv.01
ic_hex_set_edge_projection 71 72 0 1 crv.01
ic_undo_group_end
ic_undo_group_begin
ic_hex_set_edge_projection 71 74 0 1 crv.04
ic_undo_group_end
```

```
ic_undo_group_begin
ic_hex_move_node 74 GEOM.21
ic_undo_group_end
ic_undo_group_begin
ic_hex_move_node 71 GEOM.18
ic_undo_group_end
ic_undo_group_begin
ic_point crv_par GEOM pnt.08 {crv.01 0.33}
ic_undo_group_end
ic_undo_group_begin
ic_point crv_par GEOM pnt.16 {crv.01 0.67}
ic_undo_group_end
ic_undo_group_begin
ic_hex_move_node 75 pnt.08
ic_undo_group_end
ic_undo_group_begin
ic_hex_move_node 72 pnt.16
ic_undo_group_end
ic_undo_group_begin
ic_point crv_par GEOM pnt.17 {crv.07 0.67}
ic_undo_group_end
ic_undo_group_begin
ic_point crv_par GEOM pnt.18 {crv.08 0.67}
ic_undo_group_end
ic_undo_group_begin
ic_point crv_par GEOM pnt.19 {crv.07 0.33}
ic_undo_group_end
ic_undo_group_begin
ic_point crv_par GEOM pnt.20 {crv.08 0.33}
ic_undo_group_end
ic_undo_group_begin
ic_hex_set_edge_projection 21 71 0 1 crv.09
ic_undo_group_end
ic_undo_group_begin
ic_hex_set_edge_projection 25 74 0 1 crv.12
ic_undo_group_end
ic_undo_group_begin
ic_hex_move_node 21 GEOM.25
ic_undo_group_end
ic_undo_group_begin
ic_hex_move_node 25 GEOM.26
ic_undo_group_end
ic_undo_group_begin
ic_hex_set_edge_projection 25 26 0 1 crv.07
ic_hex_set_edge_projection 22 26 0 1 crv.07
ic_hex_set_edge_projection 21 22 0 1 crv.07
ic_undo_group_end
ic_undo_group_begin
ic_hex_move_node 26 pnt.19
ic_undo_group_end
ic_undo_group_begin
ic_hex_move_node 22 pnt.17
ic_undo_group_end
ic_undo_group_begin
ic_hex_set_edge_projection 37 77 0 1 crv.10
ic_undo_group_end
ic_undo_group_begin
```

```
ic_hex_set_edge_projection 41 80 0 1 crv.11
ic_undo_group_end
ic_undo_group_begin
ic_hex_move_node 77 pnt.03
ic_undo_group_end
ic_undo_group_begin
ic_hex_move_node 80 pnt.02
ic_undo_group_end
ic_undo_group_begin
ic_point intersect CRVS pnt.21 {srf.00.1e6 srf.00.1e7.0} tol 0.001
ic_undo_group_end
ic_undo_group_begin
ic_point intersect CRVS pnt.22 {srf.00.1e6 srf.00.1e7.1} tol 0.001
ic_undo_group_end
ic_undo_group_begin
ic_hex_set_edge_projection 77 78 0 1 srf.00.1e6
ic_hex_set_edge_projection 78 81 0 1 srf.00.1e6
ic_hex_set_edge_projection 80 81 0 1 srf.00.1e6
ic_undo_group_end
ic_undo_group_begin
ic_hex_move_node 78 pnt.21
ic_undo_group_end
ic_undo_group_begin
ic_hex_move_node 81 pnt.22
ic_undo_group_end
ic_undo_group_begin
ic_hex_set_edge_projection 37 38 0 1 crv.08
ic_hex_set_edge_projection 38 42 0 1 crv.08
ic_hex_set_edge_projection 41 42 0 1 crv.08
ic_undo_group_end
ic_undo_group_begin
ic_hex_move_node 38 pnt.18
ic_undo_group_end
ic_undo_group_begin
ic_hex_move_node 42 pnt.20
ic_undo_group_end
ic_undo_group_begin
ic_hex_split_grid 71 77 GEOM.15 m PNTS CRVS GEOM TEMP_TSF_FAM FLUID
ic_undo_group_end
ic_undo_group_begin
ic_hex_split_grid 96 77 CRVS.9 m PNTS CRVS GEOM TEMP_TSF_FAM FLUID
ic_undo_group_end
ic_undo_group_begin
ic_hex_mark_blocks unmark
ic_hex_mark_blocks superblock 45
ic_hex_mark_blocks superblock 46
ic_hex_change_element_id VORFN
ic_geo_delete_family TEMP_TSF_FAM
ic_undo_group_end
ic_undo_group_begin
ic_point intersect CRVS pnt.23 {srf.00e4.0 crv.02} tol 0.1
ic_undo_group_end
ic_undo
ic_geo_configure_one_attribute surface shade wire
ic_undo_group_begin
ic_undo_group_end
ic_undo_group_begin
```

```
ic_curve arc GEOM crv.03 {GEOM.16 pnt.06 PNTS.7}
ic_undo_group_end
ic_undo_group_begin
ic_undo_group_end
ic_undo_group_begin
ic_point intersect CRVS pnt.24 {srf.00e4.0 crv.03} tol 0.001
ic_undo_group_end
ic_undo_group_begin
ic_point intersect CRVS pnt.25 {srf.00e4.1 crv.03} tol 0.001
ic_undo_group_end
ic_undo_group_begin
ic_point intersect CRVS pnt.26 {srf.00e4.1 crv0} tol 0.001
ic_undo_group_end
ic_undo_group_begin
ic_point intersect CRVS pnt.26 {srf.00e4.1 srf.00.1e8} tol 0.001
ic_undo_group_end
ic_undo_group_begin
ic_point intersect CRVS pnt.27 {srf.00e4.0 srf.00.1e8} tol 0.001
ic_undo_group_end
ic_undo_group_begin
ic_hex_move_node 120 pnt99
ic_undo_group_end
ic_undo_group_begin
ic_hex_move_node 118 CRVS.9
ic_undo_group_end
ic_undo_group_begin
ic_hex_move_node 124 pnt.27
ic_undo_group_end
ic_undo_group_begin
ic_hex_move_node 126 pnt.26
ic_undo_group_end
ic_undo_group_begin
ic_hex_set_edge_projection 118 124 0 1 srf.00.1e8
ic_hex_set_edge_projection 124 126 0 1 srf.00.1e8
ic_hex_set_edge_projection 120 126 0 1 srf.00.1e8
ic_undo_group_end
ic_undo_group_begin
ic_hex_move_node 98 GEOM.16
ic_undo_group_end
ic_undo_group_begin
ic_hex_move_node 96 GEOM.15
ic_undo_group_end
ic_undo_group_begin
ic_hex_set_edge_projection 71 96 0 1 crv.06
ic_undo_group_end
ic_undo_group_begin
ic_hex_set_edge_projection 74 98 0 1 srf.05e48
ic_undo_group_end
ic_undo_group_begin
ic_hex_set_edge_projection 98 104 0 1 crv.03
ic_hex_set_edge_projection 102 104 0 1 crv.03
ic_hex_set_edge_projection 96 102 0 1 crv.03
ic_undo_group_end
ic_undo_group_begin
ic_hex_move_node 104 pnt.25
ic_undo_group_end
ic_undo_group_begin
```

```
ic_hex_move_node 102 pnt.24
ic_undo_group_end
ic_undo_group_begin
ic_hex_set_edge_projection 96 118 0 1 srf.00e4
ic_undo_group_end
ic_undo_group_begin
ic_hex_set_edge_projection 98 120 0 1 crv0
ic_undo_group_end
ic_undo_group_begin
ic_hex_set_edge_projection 104 126 0 1 srf.00e4.1
ic_undo_group_end
ic_undo_group_begin
ic_hex_set_edge_projection 102 124 0 1 srf.00e4.0
ic_undo_group_end
ic_boco_parts_dim CRVS
ic_boco_parts_dim FLUID
ic_boco_parts_dim GEOM
ic_boco_parts_dim PNTS
ic_boco_parts_dim VORFN
ic_undo_group_begin
ic_hex_project_face node_numbers { 21 22 95 101 } { 95 101 117 123 } { 37 38
117 123 } { 22 101 26 103 } { 101 123 103 125 } { 38 123 42 125 } { 25 26 97
103 } { 41 42 119 125 } { 97 103 119 125 } -to_surface srf.08
ic_undo_group_end
ic_undo_group_begin
ic_hex_project_face node_numbers { 37 77 38 78 } { 38 42 78 81 } { 41 80 42
81 } -to_surface srf.06
ic_undo_group_end
ic_undo_group_begin
ic_hex_set_edge_projection 118 77 0 1 srf.00.1e7
ic_undo_group_end
ic_undo_group_begin
ic_hex_set_edge_projection 124 78 0 1 srf.00.1e7.0
ic_undo_group_end
ic_undo_group_begin
ic_hex_set_edge_projection 126 81 0 1 srf.00.1e7.1
ic_undo_group_end
ic_undo_group_begin
ic_hex_set_edge_projection 120 80 0 1 crv.00
ic_undo_group_end
ic_undo_group_begin
ic_hex_set_mesh 118 77 n 35 h1rel 0.0 h2rel 0.0 r1 2 r2 2 lmax 0 default
copy_to_parallel unlocked
ic_undo_group_end
ic_undo_group_begin
ic_hex_set_mesh 118 77 n 35 h1rel 0.0 h2rel 0.0 r1 2 r2 2 lmax 0 default
copy_to_parallel unlocked
ic_undo_group_begin
ic_undo_group_end
ic_undo_group_end
ic_undo_group_begin
ic_hex_set_mesh 96 118 n 80 h1rel 0.0 h2rel 0.0 r1 2 r2 2 lmax 0 default
copy_to_parallel unlocked
ic_undo_group_end
ic_undo_group_begin
ic_hex_set_mesh 96 118 n 80 h1rel 0.000564569854228 h2rel 0.0 r1 1.1 r2 2
lmax 0 default copy_to_parallel unlocked
```

```
ic_undo_group_begin
ic_undo_group_end
ic_undo_group_end
ic_undo_group_begin
ic_hex_set_mesh 71 96 n 60 h1rel 0.0 h2rel 0.0 r1 2 r2 2 lmax 0 default
copy_to_parallel unlocked
ic_undo_group_end
ic_undo_group_begin
ic_hex_set_mesh 71 96 n 60 h1rel 0.0 h2rel 0.000666666666667 r1 2 r2 1.15
lmax 0 default copy_to_parallel unlocked
ic_undo_group_begin
ic_undo_group_end
ic_undo_group_end
ic_undo_group_begin
ic_hex_set_mesh 21 71 n 120 h1rel 0.0 h2rel 0.0 r1 2 r2 2 lmax 0 default
copy_to_parallel unlocked
ic_undo_group_end
ic_undo_group_begin
ic_hex_set_mesh 21 71 n 120 h1rel 0.0 h2rel 3.13269468131e-006 r1 2 r2 1.05
lmax 0 default copy_to_parallel unlocked
ic_undo_group_begin
ic_undo_group_end
ic_undo_group_end
ic_undo_group_begin
ic_geo_set_part surface srf.06 OUT 0
ic_undo_group_end
ic_undo_group_begin
ic_geo_set_part surface {srf.04 srf.03 srf.05} SYM 0
ic_undo_group_end
ic_undo_group_begin
ic_geo_set_part surface {srf.02 srf.07 srf.08} FARFIELD 0
ic_undo_group_end
ic_hex_list_family_projection
ic_hex_create_mesh PNTS CRVS GEOM FLUID OUT SYM FARFIELD proj 2 dim_to_mesh 3
ic_undo_group_begin
ic_hex_set_mesh 95 101 n 25 h1rel 0.0 h2rel 0.0 r1 2 r2 2 lmax 0 default
copy_to_parallel unlocked
ic_undo_group_end
ic_undo_group_begin
ic_hex_set_mesh 101 103 n 25 h1rel 0.0 h2rel 0.0 r1 2 r2 2 lmax 0 default
copy_to_parallel unlocked
ic_undo_group_end
ic_undo_group_begin
ic_hex_set_mesh 101 103 n 25 h1rel 0.0 h2rel 0.0 r1 2 r2 2 lmax 0 default
copy_to_parallel unlocked
ic_undo_group_begin
ic_undo_group_end
ic_undo_group_end
ic_hex_list_family_projection
ic_hex_create_mesh PNTS CRVS GEOM FLUID OUT SYM FARFIELD proj 2 dim_to_mesh 3
ic_hex_mark_blocks unmark
ic_undo_group_begin
ic_hex_mark_blocks superblock 13
ic_undo_group_end
ic_undo_group_begin
ic_hex_mark_blocks face_neighbors corners { 71 74 72 75 } { 71 96 74 98 }
{ 96 98 102 104 }
```

```
ic_undo_group_end
ic_undo_group_begin
ic_hex_ogrid 1 m PNTS CRVS GEOM FLUID OUT SYM FARFIELD -version 50
ic_hex_mark_blocks unmark
ic_undo_group_end
ic_hex_mark_blocks unmark
ic_undo_group_begin
ic_hex_set_mesh 102 146 n 8 h1rel 0.0 h2rel 0.0 r1 2 r2 2 lmax 0 default
copy_to_parallel unlocked
ic_undo_group_end
ic_undo_group_begin
ic_hex_set_mesh 102 146 n 8 h1rel 0.0 h2rel 0.0 r1 2 r2 2 lmax 0 default
copy_to_parallel unlocked
ic_undo_group_begin
ic_undo_group_end
ic_undo_group_end
ic_hex_list_family_projection
ic_hex_create_mesh PNTS CRVS GEOM FLUID OUT SYM FARFIELD proj 2 dim_to_mesh 3
ic_undo_group_begin
ic_hex_set_mesh 71 96 n 60 h1rel 0.0 h2rel 0.000666666666667 r1 2 r2 1.15
lmax 0 default copy_to_parallel unlocked
ic_undo_group_begin
ic_undo_group_end
ic_undo_group_end
ic_undo_group_begin
ic_hex_set_mesh 71 139 n 11 h1rel 0.0 h2rel 0.0 r1 2 r2 2 lmax 0 default
copy_to_parallel unlocked
ic_undo_group_end
ic_undo_group_begin
ic_hex_set_mesh 71 139 n 11 h1rel 0.0 h2rel 0.0 r1 2 r2 2 lmax 0 default
copy_to_parallel unlocked
ic_undo_group_begin
ic_undo_group_end
ic_undo_group_end
ic_undo_group_begin
ic_hex_set_mesh 139 140 n 23 h1rel 0.0 h2rel 0.0 r1 2 r2 2 lmax 0 default
copy_to_parallel unlocked
ic_undo_group_end
ic_undo_group_begin
ic_hex_set_mesh 139 140 n 23 h1rel 0.0 h2rel 0.0 r1 2 r2 2 lmax 0 default
copy_to_parallel unlocked
ic_undo_group_begin
ic_undo_group_end
ic_undo_group_end
ic_undo_group_begin
ic_hex_set_mesh 140 143 n 23 h1rel 0.0 h2rel 0.0 r1 2 r2 2 lmax 0 default
copy_to_parallel unlocked
ic_undo_group_end
ic_undo_group_begin
ic_hex_set_mesh 140 143 n 23 h1rel 0.0 h2rel 0.0 r1 2 r2 2 lmax 0 default
copy_to_parallel unlocked
ic_undo_group_begin
ic_undo_group_end
ic_undo_group_end
ic_undo_group_begin
ic_geo_set_part surface {srf.01 srf.00 srf.00.1} BODY 0
ic_undo_group_end
```

```
ic_hex_list_family_projection
ic_hex_create_mesh PNTS CRVS GEOM FLUID OUT SYM FARFIELD BODY proj 2
dim_to_mesh 3
ic_hex_write_file hex.uns PNTS CRVS GEOM FLUID OUT SYM FARFIELD BODY proj 2
dim_to_mesh 3 -family_boco family_boco.fbc
ic_uns_load hex.uns 3 0 {} 2
ic_uns_update_family_type visible {OUT FLUID GEOM FARFIELD SYM TEMP_TSF_FAM
PNTS ORFN BODY CRVS} {!NODE !LINE_2 QUAD_4 !HEXA_8} update 0
ic_uns_diag_reset_degen_min_max
ic_boco_solver
ic_uns_update_family_type visible {OUT FLUID GEOM FARFIELD SYM TEMP_TSF_FAM
PNTS ORFN BODY CRVS} {!NODE !LINE_2 QUAD_4 !HEXA_8} update 0
ic_boco_clear_icons
ic_csystem_display all 0
ic_csystem_set_current global
ic_boco_nastran_csystem reset
ic_undo_group_begin
ic_boco_solver Fluent_V6
ic_solver_mesh_info Fluent_V6
ic_undo_group_end
ic_boco_set FLUID { { 1  {color}  12109107  } }
ic_boco_set GEOM { { 1  {color}  16663866  } }
ic_boco_set OUT {{1 PRESO 0}}
ic_geo_new_family TEMP_TSF_FAM 0
ic_boco_set TEMP_TSF_FAM {}
ic_boco_set ORFN {}
ic_boco_set FARFIELD {{1 PRESF 0} { 1  {color}  16663866  }}
ic_boco_set PNTS {}
ic_boco_set SYM {{1 SYM 0}}
ic_boco_set BODY {}
ic_boco_set CRVS {}
ic_geo_new_family VORFN 0
ic_uns_new_family VORFN 1
ic_boco_set VORFN {}
ic_boco_solver
ic_boco_save F:/GAcode/SEEB-ALR_best.fbc
ic_boco_save_atr F:/GAcode/SEEB-ALR_best.atr
ic_geo_delete_family CREATED_FACES
ic_geo_delete_family TEMP_TSF_FAM
ic_save_tetin SEEB-ALR.tin 0 0 {} {} 0 0 1
ic_uns_check_duplicate_numbers
ic_save_unstruct SEEB-ALR.uns 1 {} {} {}
ic_uns_set_modified 1
ic_hex_save_blocking SEEB-ALR.blk
ic_boco_solver
ic_boco_save SEEB-ALR.fbc
ic_boco_save_atr SEEB-ALR.atr
ic_cart_is_loaded
ic_save_project_file  F:/GAcode/SEEB-ALR_best.prj  {array\  set\  file_name\
\{  {          catia_dir .} {          parts_dir .} {          domain_loaded 1}
{     cart_file_loaded 0} {     cart_file {}} {     domain_saved SEEB-ALR.uns}
{     archive {}} {     med_replay {}} {     topology_dir .} {     ugparts_dir .}
{                              icons         {{$env(ICEM_ACN)/lib/ai_env/icons}
{$env(ICEM_ACN)/lib/va/EZCAD/icons}                {$env(ICEM_ACN)/lib/icons}
{$env(ICEM_ACN)/lib/va/CABIN/icons}}}    {              tetin    SEEB-ALR.tin}
{     family_boco SEEB-ALR.fbc} {     iges_dir .} {     solver_params_loaded 0}
{     attributes_loaded 0} {     project_lock {}} {     attributes SEEB-ALR.atr}
```

{   domain SEEB-ALR.uns} {   domains_dir .} {   settings_loaded 0}
{   settings SEEB-ALR.prj} {   blocking SEEB-ALR.blk} {   hexa_replay {}}
{   transfer_dir .} {   mesh_dir .} {   family_topo {}}
{   gemsparts_dir .} {   family_boco_loaded 0} {   tetin_loaded 0}
{   project_dir .} {   topo_mulcad_out {}} {   solver_params hex.par} \}
array\ set\ options\ \{ {   expert 1} {   remote_path {}}
{   tree_disp_quad 2} {   tree_disp_pyra 0} {   evaluate_diagnostic 0}
{   histo_show_default 1} {   select_toggle_corners 0} {   remove_all 0}
{   keep_existing_file_names 0} {   record_journal 0} {   edit_wait 0}
{   face_mode 0} {   select_mode all} {   med_save_emergency_tetin 1}
{   user_name cfduser} {   diag_which all} {   uns_warn_if_display 500000}
{   bubble_delay 1000} {   external_num 1} {   tree_disp_tri 2}
{   apply_all 0} {   default_solver Fluent_V6} {   temporary_directory {}}
{   flood_select_angle 0} {   home_after_load 1} {   project_active 0}
{   histo_color_by_quality_default 1} {   undo_logging 1}
{   tree_disp_hexa 0} {   histo_solid_default 1} {   host_name CFDT3610-1}
{   xhidden_full 1} {   editor {}} {   mouse_color orange} {   clear_undo
1} {   remote_acn {}} {   remote_sh csh} {   tree_disp_penta 0}
{   n_processors 1} {   remote_host {}} {   save_to_new 0}
{   quality_info Quality} {   tree_disp_node 0}
{   med_save_emergency_mesh 1} {   redtext_color red} {   tree_disp_line 0}
{   select_edge_mode 0} {   use_dlremote 0} {   max_mesh_map_size 1024}
{   show_tris 1} {   remote_user {}} {   auto_save_views 1}
{   max_cad_map_size 512} {   display_origin 0}
{   uns_warn_user_if_display 1000000} {   detail_info 0} {   win_java_help
0} {   show_factor 1} {   boundary_mode all} {   clean_up_tmp_files 1}
{   med_save_emergency_blocking 1} {   max_binary_tetin 0}
{   tree_disp_tetra 0} \} array\ set\ disp_options\ \{ {   uns_dualmesh 0}
{   uns_warn_if_display 500000} {   uns_normals_colored 0} {   uns_icons 0}
{   uns_locked_elements 0} {   uns_shrink_npos 0} {   uns_node_type None}
{   uns_icons_normals_vol 0} {   uns_bcfield 0} {   backup Wire}
{   uns_nodes 0} {   uns_only_edges 0} {   uns_surf_bounds 0}
{   uns_wide_lines 0} {   uns_vol_bounds 0} {   uns_displ_orient Triad}
{   uns_orientation 0} {   uns_directions 0} {   uns_thickness 0}
{   uns_shell_diagnostic 0} {   uns_normals 0} {   uns_couplings 0}
{   uns_periodicity 0} {   uns_single_surfaces 0} {   uns_midside_nodes 1}
{   uns_shrink 100} {   uns_multiple_surfaces 0} {   uns_no_inner 0}
{   uns_enums 0} {   uns_disp Wire} {   uns_bcfield_name {}}
{   uns_color_by_quality 0} {   uns_changes 0} {   uns_cut_delay_count
1000} \} {set icon_size1 24} {set icon_size2 35} {set thickness_defined 0}
{set solver_type 1} {set solver_setup 1} array\ set\ prism_values\
\{ {   n_triangle_smoothing_steps 5} {   min_smoothing_steps 6}
{   first_layer_smoothing_steps 1} {   new_volume {}} {   height {}}
{   prism_height_limit {}} {   interpolate_heights 0}
{   n_tetra_smoothing_steps 10} {   do_checks {}} {   delete_standalone 1}
{   ortho_weight 0.50} {   max_aspect_ratio {}} {   ratio_max {}}
{   total_height {}} {   use_prism_v10 0} {   intermediate_write 1}
{   delete_base_triangles {}} {   ratio_multiplier {}}
{   refine_prism_boundary 1} {   max_size_ratio {}} {   triangle_quality
{}} {   max_prism_angle 180} {   tetra_smooth_limit 0.3}
{   max_jump_factor 5} {   use_existing_quad_layers 0} {   layers 3}
{   fillet 0.10} {   into_orphan 0} {   init_dir_from_prev {}}
{   blayer_2d 0} {   do_not_allow_sticking {}} {   top_family {}} {   law
exponential} {   min_smoothing_val 0.1} {   auto_reduction 0}
{   stop_columns 1} {   stair_step 1} {   smoothing_steps 12}
{   side_family {}} {   min_prism_quality 0.01} {   ratio 1.2} \} {set
aie_current_flavor {}} array\ set\ vid_options\

74

```
\{  {            wb_import_mix_res_line  0}  {          wb_import_cad_att_pre  SDFEA}
{            wb_import_surface_bodies   1}   {            wb_import_mat_points   0}
{      wb_NS_to_subset 0} {       auxiliary 0} {      wb_import_mix_res_surface 0}
{     wb_import_mix_res -1} {      wb_import_cad_att_trans 1} {     show_name 0}
{          do_intersect_self_part   1}  {            wb_import_run_mesher   tetra}
{          wb_import_delete_solids   0}  {            wb_import_solid_bodies   1}
{      wb_import_mix_res_solid 0} {       inherit 1} {       default_part GEOM}
{      new_srf_topo 1} {      DelPerFlag 0} {       wb_import_analysis_type 3}
{     wb_import_line_bodies 0} {     show_item_name 0} {      composite_tolerance
1.0}  {           wb_import_en_sym_proc   1}   {            wb_import_sel_proc   1}
{          wb_import_work_points   0}   {            wb_import_mix_res_point   0}
{       wb_import_scale_geo 0} {          replace 0} {        same_pnt_tol 1e-4}
{        tdv_axes  1} {        vid_mode  0} {        DelBlkPerFlag 0} \} {set
savedTreeVisibility   {geomNode   1   geom_subsetNode   0   geomPointNode   0
geomCurveNode 2 geomSurfNode 2 meshNode 1 mesh_subsetNode 2 meshPointNode 0
meshLineNode 0 meshShellNode 2 meshQuadNode 2 meshVolumeNode 0 meshHexaNode 0
blockingNode  1  block_subsetNode  2  block_vertNode  0  block_edgeNode  2
block_faceNode 0 block_blockNode 0 block_meshNode 0 topoNode 2 topo-root 2
partNode 2 part-BODY 2 part-CRVS 2 part-FARFIELD 2 part-FLUID 2 part-GEOM 2
part-OUT  2  part-PNTS  2  part-SYM  2  part-VORFN  0}}  {set  last_view  {rot
{0.49664128607   -0.284049121047   -0.349119049226   0.742145147037}   scale
{3.07013855914  3.07013855914  3.07013855914}  center  {71.915767580035975  -
67.856124632927077 17.107683500467129} pos {45.0773293088 -130.052118837 0}}}
array\  set\  cut_info\  \{  {          active  0}  \}  array\  set\  hex_option\
\{   {            default_bunching_ratio   2.0}   {            floating_grid   0}
{       n_tetra_smoothing_steps  20} {        trfDeg  1} {        wr_hexa7  0}
{      smooth_ogrid  0} {       find_worst 1-3} {        hexa_verbose_mode  0}
{         old_eparams   0}   {             uns_face_mesh_method   uniform_quad}
{      multigrid_level 0} {      uns_face_mesh one_tri} {       check_blck 0}
{        proj_limit  0} {        check_inv  0} {        project_bspline  0}
{       hexa_update_mode 1} {       default_bunching_law BiGeometric} \} array\
set\ saved_views\ \{ {      views {}} \}} {ICEM CFD}
ic_exec   {C:/Program   Files/ANSYS   Inc/v145/icemcfd/win64_amd/icemcfd/output-
interfaces/fluent6} -dom {F:/GAcode/SEEB-ALR_best.uns} -b SEEB-ALR_best.fbc -
dim2d ./SEEB-ALR_best
ic_uns_num_couplings
ic_undo_group_begin
ic_uns_create_diagnostic_edgelist 1
ic_uns_diagnostic   subset   all   diag_type   uncovered   fix_fam   FIX_UNCOVERED
diag_verb {Uncovered faces} fams {} busy_off 1 quiet 1
ic_uns_create_diagnostic_edgelist 0
ic_undo_group_end
exit
```

Appendix I

FLUENT Journal File

(cx-gui-do cx-activate-item "MenuBar*ReadSubMenu*Mesh...")

(cx-gui-do cx-set-text-entry "Select File*FilterText" "f:\gacode*")

(cx-gui-do cx-activate-item "Select File*Apply")

(cx-gui-do cx-set-text-entry "Select File*Text" "SEEB-ALR.msh")

(cx-gui-do cx-activate-item "Select File*OK")

(cx-gui-do   cx-activate-item
"General*Frame1*Table1*Frame1(Mesh)*ButtonBox1(Mesh)*PushButton1(Scale)")

(cx-gui-do   cx-set-list-selections   "Scale
Mesh*Frame2(Scaling)*Table2(Scaling)*DropDownList2(Mesh Was Created In)" '(4))

(cx-gui-do cx-activate-item "Scale Mesh*Frame2(Scaling)*Table2(Scaling)*DropDownList2(Mesh
Was Created In)")

(cx-gui-do cx-activate-item "Scale Mesh*Frame2(Scaling)*Table2(Scaling)*PushButton4(Scale)")

(cx-gui-do cx-activate-item "Scale Mesh*PanelButtons*PushButton1(Close)")

(cx-gui-do   cx-set-toggle-button
"General*Frame1*Table1*Frame2(Solver)*Table2(Solver)*ButtonBox1(Type)*Density-Based" #f)

(cx-gui-do   cx-activate-item
"General*Frame1*Table1*Frame2(Solver)*Table2(Solver)*ButtonBox1(Type)*Density-Based")

(cx-gui-do cx-activate-item "NavigationPane*Frame1*PushButton5(Models)")

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

(cx-gui-do cx-activate-item "Models*Frame1*Table1*Frame1*List1(Models)")

(cx-gui-do cx-activate-item "Models*Frame1*Table1*PushButton2(Edit)")

(cx-gui-do   cx-set-toggle-button
"Energy*Frame1(Energy)*Table1(Energy)*Frame1*ToggleBox1*CheckButton1(Energy   Equation)"
#f)

(cx-gui-do   cx-activate-item
"Energy*Frame1(Energy)*Table1(Energy)*Frame1*ToggleBox1*CheckButton1(Energy Equation)")

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

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

(cx-gui-do cx-activate-item "Models*Frame1*Table1*Frame1*List1(Models)")

(cx-gui-do cx-activate-item "Models*Frame1*Table1*PushButton2(Edit)")

(cx-gui-do   cx-set-toggle-button   "Viscous
Model*Frame1*Table1*Frame1(Model)*ToggleBox1(Model)*Spalart-Allmaras (1 eqn)" #f)

(cx-gui-do   cx-activate-item   "Viscous
Model*Frame1*Table1*Frame1(Model)*ToggleBox1(Model)*Spalart-Allmaras (1 eqn)")

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

(cx-gui-do cx-activate-item "NavigationPane*Frame1*PushButton6(Materials)")

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

(cx-gui-do cx-activate-item "Materials*Frame1*Table1*Frame1*List1(Materials)")

(cx-gui-do cx-activate-item "Materials*Frame1*Table1*ButtonBox2*PushButton1(Create/Edit)")

(cx-gui-do   cx-set-list-selections   "Create/Edit
Materials*Frame2(Properties)*Table2(Properties)*Frame4*Frame2*DropDownList1" '(1))

(cx-gui-do   cx-activate-item   "Create/Edit
Materials*Frame2(Properties)*Table2(Properties)*Frame4*Frame2*DropDownList1")

(cx-gui-do cx-activate-item "Create/Edit Materials*PanelButtons*PushButton1(Change/Create)")

(cx-gui-do cx-activate-item "Create/Edit Materials*PanelButtons*PushButton1(Close)")

(cx-gui-do cx-activate-item "NavigationPane*Frame1*PushButton8(Cell Zone Conditions)")

(cx-gui-do cx-activate-item "Boundary Conditions*Frame1*Table1*Frame2*Table2*Frame4*Table4*Frame2*Table2*PushButton2(Operating Conditions)")

(cx-gui-do cx-set-real-entry-list "Operating Conditions*Frame1*Frame1(Pressure)*Table1(Pressure)*RealEntry2(Operating Pressure)" '(0))

(cx-gui-do cx-activate-item "Operating Conditions*PanelButtons*PushButton1(OK)")

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

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

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

(cx-gui-do cx-activate-item "Boundary Conditions*Frame1*Table1*Frame2*Table2*Frame4*Table4*ButtonBox1*PushButton1(Edit)")

(cx-gui-do cx-set-real-entry-list "pressure-far-field-18-1*Frame4*Frame1(Momentum)*Frame1*Table1*Frame12*Table12*RealEntry2(Mach Number)" '(1.6))

(cx-gui-do cx-set-real-entry-list "pressure-far-field-18-1*Frame4*Frame1(Momentum)*Frame1*Table1*Frame11*Table11*RealEntry2(Gauge Pressure)" '(101325))

(cx-gui-do cx-set-real-entry-list "pressure-far-field-18-1*Frame4*Frame1(Momentum)*Frame1*Table1*Frame3(Turbulence)*Table3(Turbulence)*Frame13*Table13*RealEntry2(Turbulent Viscosity Ratio)" '(5))

(cx-gui-do cx-activate-item "pressure-far-field-18-1*PanelButtons*PushButton1(OK)")

(cx-gui-do cx-set-list-selections "Boundary Conditions*Frame1*Table1*Frame1*List1(Zone)" '(3))

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

(cx-gui-do cx-activate-item "Boundary Conditions*Frame1*Table1*Frame2*Table2*Frame4*Table4*ButtonBox1*PushButton1(Edit)")

(cx-gui-do cx-set-real-entry-list "pressure-outlet-16-1*Frame4*Frame1(Momentum)*Frame1*Table1*Frame4*Table4*RealEntry2(Gauge Pressure)" '(101325))

(cx-gui-do cx-set-real-entry-list "pressure-outlet-16-1*Frame4*Frame1(Momentum)*Frame1*Table1*Frame3(Turbulence)*Table3(Turbulence)*Frame1 3*Table13*RealEntry2(Backflow Turbulent Viscosity Ratio)" '(5))

(cx-gui-do cx-activate-item "pressure-outlet-16-1*PanelButtons*PushButton1(OK)")

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

(cx-gui-do cx-set-list-selections "Reference Values*DropDownList1(Compute from)" '(4))

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

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

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

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

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

(cx-gui-do cx-activate-item "Solution Initialization*Frame1*Table1*ButtonBox10*PushButton2(Initialize)")

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

(cx-gui-do cx-set-toggle-button "Run Calculation*Frame1*Table1*CheckButton16(Solution Steering)" #f)

(cx-gui-do cx-activate-item "Run Calculation*Frame1*Table1*CheckButton16(Solution Steering)")

(cx-gui-do cx-set-list-selections "Run Calculation*Frame1*Table1*Frame18*Table18*DropDownList1(Flow Type)" '(3))

(cx-gui-do cx-activate-item "Run Calculation*Frame1*Table1*Frame18*Table18*DropDownList1(Flow Type)")

(cx-gui-do cx-activate-item "Run Calculation*Frame1*Table1*Frame18*Table18*PushButton6(More Settings)")

(cx-gui-do cx-set-real-entry-list "Solution Steering*Frame1*Frame1(Steering Settings)*Frame1*Table1*Frame2(Courant Number)*Table2(Courant Number)*RealEntry1(Initial)" '(0.1))

(cx-gui-do cx-set-real-entry-list "Solution Steering*Frame1*Frame1(Steering Settings)*Frame1*Table1*Frame2(Courant Number)*Table2(Courant Number)*RealEntry2(Maximum)" '(5))

(cx-gui-do cx-activate-item "Solution Steering*PanelButtons*PushButton1(OK)")

(cx-gui-do cx-set-integer-entry "Run Calculation*Frame1*Table1*IntegerEntry9(Number of Iterations)" 2200)

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

(cx-gui-do cx-activate-item "NavigationPane*Frame1*PushButton21(Graphics and Animations)")

(cx-gui-do cx-activate-item "ToolBar*General Tools*savepicture")

(cx-gui-do cx-activate-item "Save Picture*PanelButtons*PushButton1(OK)")

(cx-gui-do cx-set-text-entry "Select File*FilterText" "f:\gacode*")

(cx-gui-do cx-activate-item "Select File*Apply")

(cx-gui-do cx-set-text-entry "Select File*Text" "FluentRunCheck.jpg")

(cx-gui-do cx-activate-item "Select File*OK")

(cx-gui-do cx-activate-item "Save Picture*PanelButtons*PushButton2(Cancel)")

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

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

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

(cx-gui-do cx-activate-item "NavigationPane*Frame1*PushButton4(General)")

(cx-gui-do   cx-activate-item

"General*Frame1*Table1*Frame1(Mesh)*ButtonBox1(Mesh)*PushButton2(Display)")

(cx-gui-do cx-activate-item "Mesh Display*Frame2*PushButton2(New Surface)")

(cx-gui-do cx-activate-item "MenuBar*PopupMenuNewSurface*Line/Rake...")

(cx-gui-do   cx-set-real-entry-list   "Line/Rake   Surface*Frame4(End

Points)*Frame1*Table1*RealEntry1(x0)" '(0.6))

(cx-gui-do   cx-set-real-entry-list   "Line/Rake   Surface*Frame4(End

Points)*Frame1*Table1*RealEntry4(x1)" '(1.2))

(cx-gui-do   cx-set-real-entry-list   "Line/Rake   Surface*Frame4(End

Points)*Frame1*Table1*RealEntry2(y0)" '(0))

(cx-gui-do   cx-set-real-entry-list   "Line/Rake   Surface*Frame4(End

Points)*Frame1*Table1*RealEntry5(y1)" '(0))

(cx-gui-do   cx-set-real-entry-list   "Line/Rake   Surface*Frame4(End

Points)*Frame1*Table1*RealEntry3(z0)" '(0.538))

(cx-gui-do   cx-set-real-entry-list   "Line/Rake   Surface*Frame4(End

Points)*Frame1*Table1*RealEntry6(z1)" '(0.538))

(cx-gui-do cx-set-text-entry "Line/Rake Surface*TextEntry6(New Surface Name)" "h/l=1.2")

(cx-gui-do cx-activate-item "Line/Rake Surface*PanelButtons*PushButton1(OK)")

(cx-gui-do cx-activate-item "Line/Rake Surface*PanelButtons*PushButton2(Cancel)")

```
(cx-gui-do cx-activate-item "Mesh Display*PanelButtons*PushButton2(Cancel)")

(cx-gui-do cx-activate-item "NavigationPane*Frame1*PushButton22(Plots)")

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

(cx-gui-do cx-activate-item "Plots*Frame1*Table1*Frame1*List1(Plots)")

(cx-gui-do cx-activate-item "Plots*Frame1*Table1*PushButton2(Set Up)")

(cx-gui-do cx-set-list-selections "Solution XY Plot*Frame9*Frame1*List1(Surfaces)" '( 2))

(cx-gui-do cx-activate-item "Solution XY Plot*Frame9*Frame1*List1(Surfaces)")

(cx-gui-do        cx-set-toggle-button        "Solution        XY
Plot*Frame1(Options)*ToggleBox1(Options)*CheckButton4(Write to File)" #f)

(cx-gui-do        cx-activate-item        "Solution        XY
Plot*Frame1(Options)*ToggleBox1(Options)*CheckButton4(Write to File)")

(cx-gui-do cx-activate-item "Solution XY Plot*PanelButtons*PushButton1(OK)")

(cx-gui-do cx-set-text-entry "Select File*Text" "FLUENT_DES_VAR_1")

(cx-gui-do cx-activate-item "Select File*OK")

(cx-gui-do cx-activate-item "Solution XY Plot*PanelButtons*PushButton2(Cancel)")

(cx-gui-do cx-activate-item "ToolBar*General Tools*savepicture")

(cx-gui-do cx-activate-item "Save Picture*PanelButtons*PushButton1(OK)")

(cx-gui-do cx-set-text-entry "Select File*Text" "FluentDone.jpg")

(cx-gui-do cx-activate-item "Select File*OK")

(cx-gui-do cx-activate-item "Save Picture*PanelButtons*PushButton2(Cancel)")

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

exit

o
```

Vita

Junhui Li

Degrees	M.S. Aerospace Engineering, Washington University in St.Louis, USA, Aug 2016
	B.S. Aircraft Design and Engineering, Beihang University, China, June 2014
Birth of Place	Taiyuan, Shanxi, China
Publications	Junhui Li, Tim Wray, and Ramesh K. Agarwal, "Shape Optimization of Supersonic Bodies to Reduce Sonic Boom Signature," AIAA Paper 2016-3432, 34th AIAA Applied Aerodynamics Conference, Washington, D.C., 13-17 June 2016.

Aug 2016