Shape Optimization of Joukowski Airfoils in Supersonic Flows

ME 7080 Final Project

Koorosh Gobal

Scope of Work:

Conformal mapping is a mathematical technique used to convert (or map) one mathematical problem and solution into another.

Many years ago, the Russian mathematician Joukowski developed a mapping function that converts a circular cylinder into a family of airfoil shapes. If points in the cylinder plane are represented by the complex coordinates x for the horizontal and y for the vertical, then every point z is specified by:

$$z=x+iy$$

Similarly, in the airfoil plane, the horizontal coordinate is B and the vertical coordinate is C, and every point A is specified by:

$$A=B+iC$$

Then Joukowski's mapping function that relates points in the airfoil plane to points in the cylinder plane is given as:

$$A=\left(z+z'\right)+\frac{b^2}{z+z'} \tag{1}$$

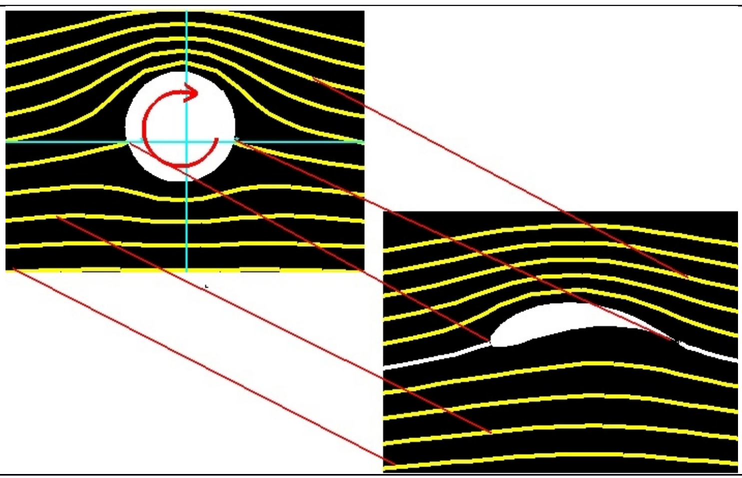


Figure 1. Joukowski transformation

The mapping function also converts the entire flow field around the cylinder into the flow field around the airfoil. We know the velocity and pressures in the plane containing the cylinder. The mapping function gives us the velocity and pressures around the airfoil. Knowing the pressure around the airfoil, we can then compute the lift. The computations are difficult to perform by hand, but can be solved quickly on a computer. [1]

In this problem the Joukowski's mapping function was only used to create the airfoil shape based on different shape parameters. These airfoils were then solved in supersonic flows. The need for such a design arises when an aircraft is required to operate consistently in the supersonic flight regime.

Supersonic airfoils generally have a thin section formed of either angled planes or opposed arcs, with very sharp leading and trailing edges. The sharp edges prevent the formation of a detached bow shock in front of the airfoil as it moves through the air [2]. This shape is in contrast to subsonic airfoils, which often have rounded leading edges to reduce flow separation over a wide range of angle of attack [3].

It is also worth mentioning that conformal mapping can be used to map the pressure over a rotating cylinder to this family of airfoils [4][5] and thus calculating lift and drag on them based on single CFD run on rotating cylinder. In that case some other assumptions like potential flow will apply. The approach used here was to explicitly solve flow over each different airfoil using CFD code rather than use of conformal mapping. Conformal mapping can be done in future works.

Design Goals:

As in other airfoil optimization cases, here the goal is to *minimize the drag coefficient* while *maintaining a lift and pitching coefficient*. Another constraint for *gliding ratio* was added to the problem formulation. Furthermore side bounds were applied on geometric features of airfoil to maintain its feasibility. In the following lines the reason for selecting different constraints are discussed.

It is clear that <u>drag coefficient</u> is needed to be minimized so airfoil would have better performance but at the same time it needs to generate enough lift to satisfy its duty to generate lift for airplane.

The <u>pitching moment coefficient</u> is important in the study of the longitudinal static stability of aircraft and missiles. The pitching moment coefficient is defined as follows

$C_m = \frac{M}{qSc}$	(2)

where M is the pitching moment, q is the dynamic pressure, S is the platform area, and c is the length of the chord of the airfoil. Pitching moment coefficient is fundamental to the definition of aerodynamic center of an airfoil. The aerodynamic center is defined to be the point on the chord line of the airfoil at which the pitching moment coefficient does not vary with angle of attack or at least does not vary significantly over the operating range of angle of attack of the airfoil. The aerodynamic center of an airfoil is usually close to 25% of the chord behind the leading edge of the airfoil.

Pitching moment is, by convention, considered to be positive when it acts to pitch the airfoil in the nose-up direction. Conventional cambered airfoils supported at the aerodynamic center pitch nose-down so the pitching moment coefficient of these airfoils is negative [6].

In aerodynamics, the glide ratio is the amount of lift generated by a wing or vehicle, divided by the drag it creates by moving through the air. A higher or more favorable L/D ratio is typically one of the major goals in aircraft design; since a particular aircraft's required lift is set by its weight, delivering that lift with lower drag leads directly to better fuel economy, climb performance, and glide ratio.

At very high speeds, lift to drag ratios tends to be lower. Dietrich Küchemann developed an empirical relationship for predicting L/D ratio for high Mach number [7]:

$\frac{L}{D_{max}} = \frac{4(M+3)}{M}$	(3)

where M is the Mach number.

As discussed above the aim of this project is to optimize the shape of airfoil to reduce the drag and maintain other features like lift coefficient and etc. The next step in formulating the problem is to optimize the airfoil performance in range of Mach numbers.

The airfoil will undergo different flight conditions through its life so it is needed to be optimized according to different fight conditions. Here the optimization performed under three different Mach numbers. To do so the cost functions (that is drag coefficient) was added together using weights. This cost function was optimized using constraint in three different flight conditions.

Problem Formulation:

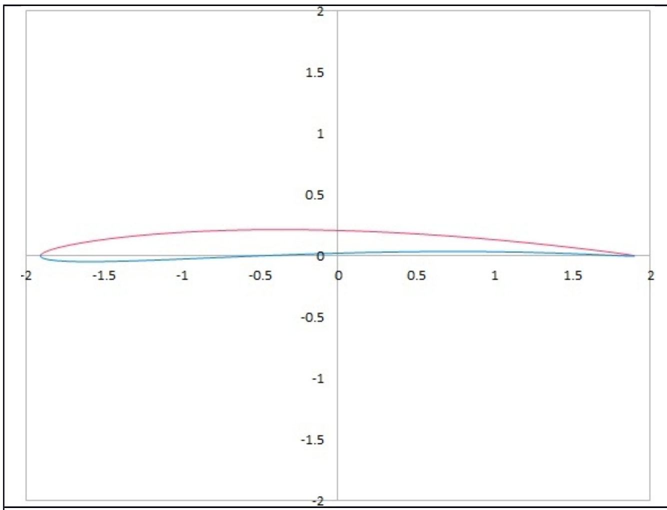
Creating the geometry:

As said in previous sections the geometry was created with Joukowski's mapping function. The equation for defining airfoil shape is derived as shown below.

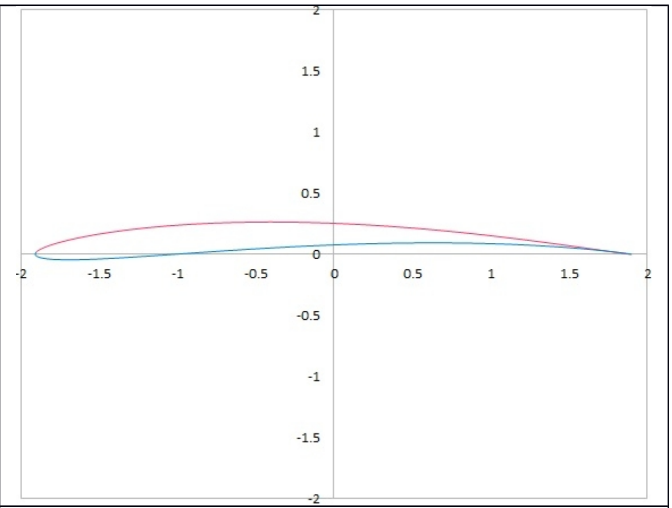
$$x = R\cos\theta \,,\, y = R\sin\theta \rightarrow \begin{cases} z = R\,e^{i\theta} \\ z' = r\,e^{i\phi} \end{cases} \rightarrow Z = \left(z + z'\right) + \frac{b^2}{z + z'}$$

$$\rightarrow \begin{cases} X = \left(R Cos\theta + r Cos\phi\right)\left(1 + \beta\right) \\ Y = \left(R Sin\theta + r Sin\phi\right)\left(1 - \beta\right) \end{cases} \wedge \beta = \frac{b^2}{\left(R Cos\theta + r Cos\phi\right)^2 + \left(R Sin\theta + r Sin\phi\right)^2} \qquad (4)$$

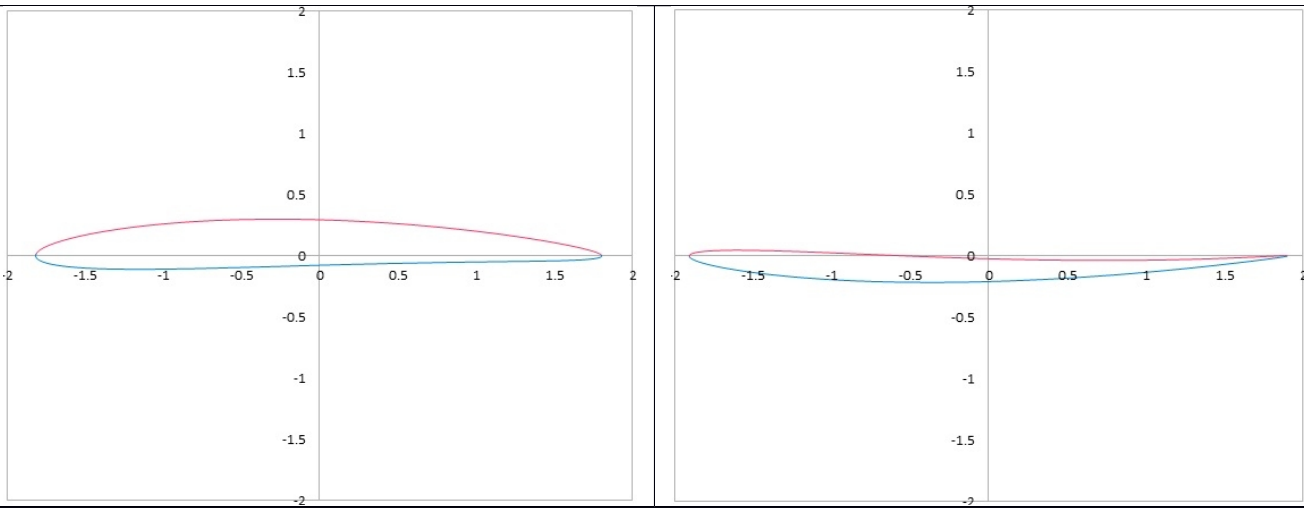
In above formulation X and Y are the coordinates of airfoil after transformation. As showed above the final airfoil is created based on initial circle defined by x and y. To change the shape of airfoil ϕ , r and b were changed to change the shape. R value is fixed and θ varies from 0 The role of each variable in changing the shape is shown in figure 2.



r = 0.07 – b = 0.95 – phi = 120



r = 0.1 – b = 0.95 – phi = 120



r = 0.07 – b = 0.90 – phi = 120	r = 0.07 – b = 0.95 – phi = 240

Figure 2. Airfoil geometry

Formulation derived in equation (4) was implemented in a excel sheet so points creating the airfoil shape can be easily modified by changing the values of ϕ , r and b. These point were later imported to mesh generator software to create the CFD mesh.

CFD Solver:

In order to solve the optimization problem the first step is to solve the supersonic flow around the airfoil and calculate desired feature of airfoil performance. To do so OpenFOAM was selected as CFD solver.

OpenFOAM (Open Source Field Operation and Manipulation) is a C++ toolbox for the development of customized numerical solvers, and pre-/post-processing utilities for the solution of continuum mechanics problems, including computational fluid dynamics (CFD). The code is based on Linux operating system and is free and open source software. An extensive set of OpenFOAM solvers has evolved that are available to users. OpenFOAM is used mainly for CFD but has found use in other areas such as stress analysis, electromagnetics and finance because it is fundamentally a tool for solving partial differential equations rather than a CFD package in the traditional sense.

OpenFOAM compressible solvers were validated by two different test cases. Validation cases were *oblique shock on a 15 degree wedge at Mach 2.5* and *shock tube* test. These problems were selected because they have same characteristics as out problem and also because their analytical results are available. Results of validation cases are available in appendix 1.

OpenFOAM has different solvers for solving the compressible supersonic flow, among them and according to validation cases the `sonicFoam` solver was selected as compressible flow solver. `sonicFoam` is a solver for transsonic/supersonic, laminar or turbulent flow of a compressible gas. Cases were setup in OpenFOAM by defining the geometry, mesh, boundary condition and solver setting.

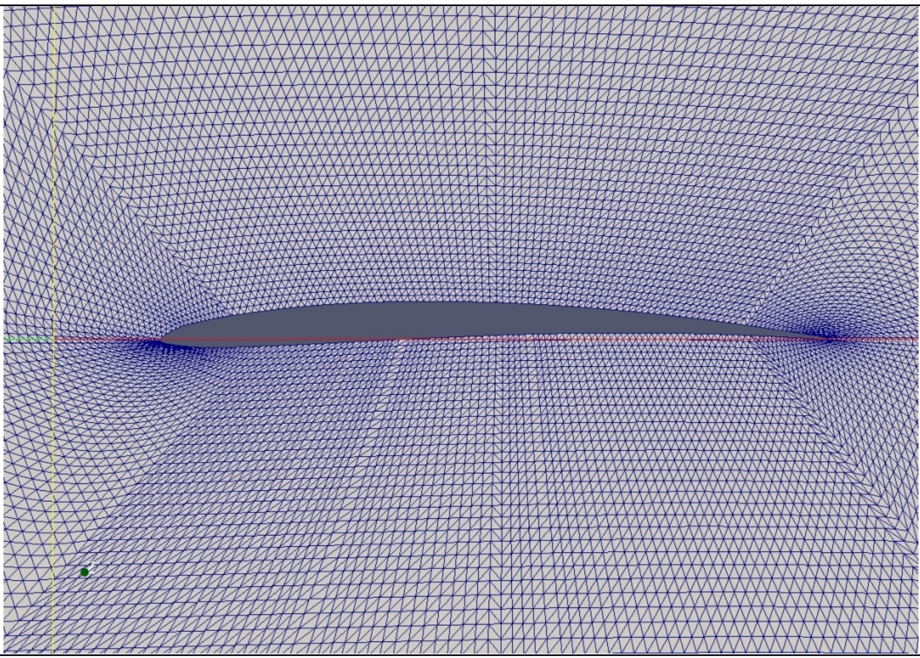
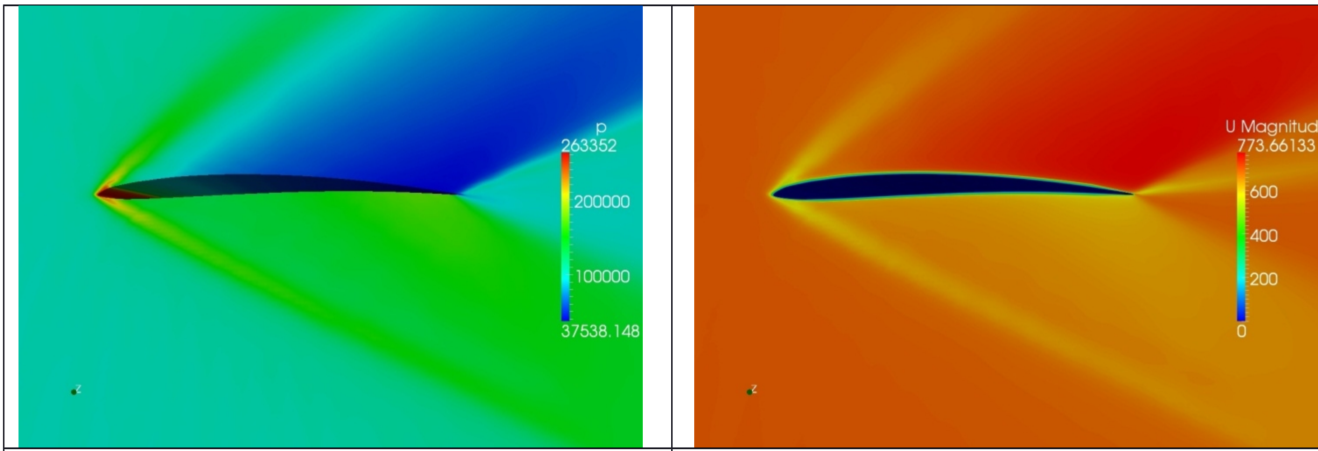


Figure 3. Mesh

The boundary conditions were applied as velocity vectors at inlet and upper and lower walls. The `zerogradient` velocity was assigned at the outlet. On the airfoil the no slip condition was assumed. Atmospheric pressure was defined at inlet and also a `zerogradient` condition was applied at the upper and lower walls for pressure. Since we have bow shocks happening on the airfoil special boundary conditions need to be applied at the outlet. `waveTransmissive` boundary condition was applied for pressure at outlet because other boundary conditions would reflect the shockwave as it reached them. On the airfoil the `zerogradient` boundary condition was applied. The turbulence model used in this simulation was k - ϵ with near wall corrections.

The each new geometry was meshed with OpenFOAM meshing code, `blockMesh`, with hexahedral meshes. Total mesh used was about 375,000 for each case. Run time for each case was about six hours on an Intel core 2 duo with 3.2GHz CPU with 4GB of RAM.



Counters of pressure | Counters of velocity

Figure 4. results for Mach 2.0 case

The lift, drag and pitching coefficients are needed values that should be exported from the CFD results. In order to get the required coefficients the case file in OpenFOAM was modified so it would export to require data to an external text file.

OpenFOAM has a built in library (`libforces.so`) that can be used for exporting forces and force coefficients from the solver. To do so the following lines was added to the controlDic directory of case file.

```
type                    forceCoeffs;
functionObjectLibs      ( "libforces.so" );
outputControl           timeStep;
outputInterval          500;

patches                 (airFoilWall);
pName                   p;
UName                   U;
rhoName                 rhoInf;
log                     true;

liftDir                 (0 1 0);
dragDir                 (1 0 0);
CofR                    (-0.9 0 0);
pitchAxis               (0 0 -1);

magUInf                 695.4619;
rhoInf                  1.1644;
```

Above code would export the force coefficients in every 500 time steps on the surface defined in `patches` section. The lift and drag coefficient were defined as shown above in `liftDir` and `dragDir` sections. The `CofR` is the aerodynamic center that is used for pitching coefficient. This point is selected to be close to 25% of the chord behind the leading edge of the airfoil. Also the pitching axis was selected so the coordinate system used here would match the one used by airfoil community.

Function Approximation:

The way optimization works is to start from a point and try to find a direction in design space to reduce the cost function value. To do that, the optimization algorithm needs to call the CFD box each time it modifies the geometry to calculate the cost function. Since each CFD simulation takes six hours to complete this approach is not feasible.

To reduce the cost of CFD two methods can be used. In the first method as stated before we can use the Joukowski's conformal mapping to map the flow around a rotating cylinder to an airfoil and the used this mapping as a function approximation for different airfoil shapes. Certain assumption will rise for this method but we can calculate sensitivities and build a good function approximation for this method. Also the total calls to the CFD solver will reduce to one.

The other method that is used here is to build a response surface based on limited experiments to represent the response of system to certain geometric changes. The critical point is where to conduct experiments in order to get the best results.

As stated in equation (4) we have three design variables that define the shape of the airfoil. Those are ϕ, r and b. Also angle of attack, α, is selected as another design variable. So there is the total of four design variables. The central composite design was used as the method to design the experiments. The Matlab function `ccdesign(n)` with n value as number of design variables was used to create the experiment table. The experiment talbe is calculated as shown in table 1.

Case number	r	ϕ	b	α	Case number	r	ϕ	b	α
1	-1	-1	-1	-1	13	1	1	-1	-1
2	-1	-1	-1	1	14	1	1	-1	1
3	-1	-1	1	-1	15	1	1	1	-1
4	-1	-1	1	1	16	1	1	1	1
5	-1	1	-1	-1	17	-2	0	0	0
6	-1	1	-1	1	18	2	0	0	0
7	-1	1	1	-1	19	0	-2	0	0
8	-1	1	1	1	20	0	2	0	0
9	1	-1	-1	-1	21	0	0	-2	0
10	1	-1	-1	1	22	0	0	2	0
11	1	-1	1	-1	23	0	0	0	-2
12	1	-1	1	1	24	0	0	0	2

Table 1. Experiments

Basically the above table numbers state the values of design variables for every experiment. The value -2 corresponds to the lowest limit of that variable and +2 to its upper limit.

The CFD simulation was run based on values specified by table 1 for three different Mach numbers 1.2, 1.4 and 2.0. Lift, drag and pitching coefficients were calculated by CFD solver. The values are attached to appendix 2.

The response surface with interaction terms included was built based on experiments conducted for different Mach numbers. Three different response surfaces were built for lift, drag and pitching coefficients. The coefficients of response surfaces are listed in table 2. The superscripts specify the Mach number that the experiment was performed. The response surface is formulated as:

$$Y=\beta_0+\beta_1 r+\beta_2\phi+\beta_3 b+\beta_4\alpha+\beta_5 r\phi+\beta_6 rb+\beta_7 r\alpha+\beta_8\phi b+\beta_9\phi\alpha+\beta_{10}b\alpha \qquad (5)$$

	β_0	β_1	β_2	β_3	β_4	β_5	β_6	β_7	β_8	β_9	β_{10}
$C_d^{(1.2)}$	14.08	-34.97	-0.01	-15.33	0.12	0.01	52.17	-0.51	0.01	0.00	-0.18
$C_l^{(1.2)}$	-74.11	171.52	0.29	75.20	-1.94	-0.22	-145.75	1.17	-0.28	0.00	3.41
$C_m^{(1.2)}$	-19.31	148.82	0.01	23.76	0.35	-0.12	-139.61	-0.39	-0.02	0.00	-0.54
$C_d^{(1.4)}$	18.68	-54.35	0.00	-20.77	0.00	-0.01	79.79	-0.14	0.00	0.00	-0.07
$C_l^{(1.4)}$	-105.71	251.47	0.37	105.00	1.50	-0.27	-217.67	-1.05	-0.34	0.00	0.12
$C_m^{(1.4)}$	-33.65	154.24	0.09	38.33	0.15	-0.18	-134.29	0.06	-0.10	0.00	-0.26
$C_d^{(2.0)}$	32.26	-43.70	0.02	-35.12	-0.18	-0.01	72.99	-0.07	-0.02	0.00	0.07
$C_l^{(2.0)}$	-162.27	393.65	0.50	165.83	4.42	-0.12	-397.25	-3.47	-0.48	0.00	-2.54
$C_m^{(2.0)}$	-26.05	125.95	0.10	29.84	-0.78	-0.24	-92.83	0.59	-0.11	0.00	0.72

Table 2. Response surface coefficients

By comparing the coefficients in table 2 and response surface equation it can be concluded that ϕ , α and the term r×b have the most contribution on the response. The impact of changing ϕ , r and b are shown on figure 2.

To check how well the regression line fits a set of data the coefficient of determination R^2 was calculated for each for response surfaces specified above. An R^2 near 1.0 indicates that a regression line fits the data well, while an R^2 closer to 0 indicates a regression line does not fit the data very well. Table 3 shows the R^2 value of each response surface.

Response surface	$C_d^{(1.2)}$	$C_l^{(1.2)}$	$C_m^{(1.2)}$	$C_d^{(1.4)}$	$C_l^{(1.4)}$	$C_m^{(1.4)}$	$C_d^{(2.0)}$	$C_l^{(2.0)}$	$C_m^{(2.0)}$
R^2	0.97306	0.99061	0.99416	0.99797	0.96818	0.99867	0.99835	0.93252	0.99934

Table 3. coefficient of determination for different Response surface coefficients

As shown in table 3 the coefficient of determination is near 1.0 for most of response surfaces. So it can be concluded that the response surface fit the data to a good extend.

Problem Statement:

Equation (4) is used for creating geometry. The design variables are ϕ , r and b as stated before. These were used to modify the geometry and create new shapes. R is set equal to one and θ changes from 0 to 2 π . The other design variable was selected as the angle of attack of the airfoil.

Side bounds needs to be applied to on shape so it meets the geometry constraints on the airfoil based on [8] and stated as below. The percentages are from camber line.

20% ≤ Maximum Camber Location ≤ %80
7% ≤ Maximum Thickness ≤ %20
0° ≤ Angle of Attack ≤ 12°

These can be turned into the limits on design variables ϕ , r and b. It is also needed to have a constraint to check for self-intersection of the shape. This can be done by checking if the upper surface is always on top of the lower surface. This wall implemented as a function in Matlab, `geoCheck` that will return zero if the geometry is not intersecting and -1 if it is intersecting. This can be implemented to optimization solver as zero equality constraint. Angle of attack α is chosen to be between zero and twelve.

The optimization problem can be formulated as

$$min f(x) = w_1 C_d^{(1.2)} + w_2 C_d^{(1.4)} + w_1 C_d^{(2.0)}$$

$C_m^{(1.2)} \leftarrow 1.0$

$C_m^{(1.4)} \leftarrow 1.0$

$C_m^{(2.0)} \leftarrow 1.0$

$C_l^{(1.2)} > 6.0$

$C_l^{(1.4)} > 6.0$

$C_l^{(2.0)} > 6.0$

$C_l^{(1.2)} / C_d^{(1.2)} > 9.0$

$C_l^{(1.4)} / C_d^{(1.4)} > 9.0$

$C_l^{(2.0)} / C_d^{(2.0)} > 9.0$

0.07 ≤ r ≤ 0.1875

120 ≤ ϕ ≤ 250

0.875 ≤ b ≤ 0.95

0 ≤ α ≤ 12

Check for shape to be not intersecting

The weights w_i are selected by procedures same as *expert's opinion*. The weights were selected as the plane travels 30% of its life at Mach 1.2, 40% of its life at Mach 1.4 and 40% of its life time at Mach 2.0.

The upper value for pitching moment was selected as to maintain the stability of the airfoil. The gliding ratio was selected from equation (3) as 10.0. Also the lift coefficient was selected to be more than 6.0.

Solution Approach:

The response surfaces built in problem formulation was used in optimization process. The response surface for C_d was implemented with weight functions for the cost function. The constraints were also implemented using response surfaces from previous section.

The optimization was used using `fmincon` to solve the problem defined in previous section. The algorithm used was sequential quadratic programming. The optimization solution of the problem is showed in table 4. Also the predicted values from response surfaces are shown in table 5.

r	ϕ	b	α
0.07	120	0.95	7.548

Table 4. Optimization result

C_l			C_d			C_m			C_l / C_d		
1.2	1.4	2.0	1.2	1.4	2.0	1.2	1.4	2.0	1.2	1.4	2.0
8.3807	8.2185	10.057	0.26549	0.18164	0.38683	1	1.3266	1.314	31.566	45.247	26

Table 5. Response surface results at optimum point

The shape defined by parameters in table 4 was created in OpenFOAM and solved to find the force and moment coefficients acting on the airfoil. The result of CFD simulation based on geometry defined in table 4 is shown in table 6.

C_l			C_d			C_m			C_l / C_d		
1.2	1.4	2.0	1.2	1.4	2.0	1.2	1.4	2.0	1.2	1.4	2.0
8.20		8.54	0.63		0.93	1.05		1.19	13.01		9.18

Table 6. CFD Results

The results of table 6 were added to response surface to create a new one and optimization was run based on new response surface. The results were the same.

Apendix 1: Validation cases

Apendix 2: Design of experiment results

References:

1. http://www.grc.nasa.gov/WWW/k-12/airplane/map.html
2. Courant & Friedrichs. Supersonic Flow and Shock Waves. Vol I. New York: Inter science Publishers, inc, 1948.
3. Zucker, Robert & Biblarz, Oscar. Fundamentals of Gas Dynamics. Second Edition.ISBN 0-471-05967-6 John Wiley & Sons, Inc.
4. Nitin R. Kapania, Katherine Terracciano, Shannon Taylor. Modeling the Fluid Flow around Airfoils Using Conformal Mapping. August 29, 2008.
5. Aravind Bhimarasetty And Raghuraman N Govardhan. A simple method for potential flow simulation of cascades. Sadhana Vol. 35, Part 6, December 2010, pp. 649–657. Indian Academy of Sciences.
6. Ira H. Abbott, and Albert E. Von Doenhoff (1959), Theory of Wing Sections, Dover Publications Inc., New York SBN 486-60586-8.
7. Hypersonic Vehicle Design. http://www.aerospaceweb.org/design/waverider/design.shtml
8. Christian Wauquiez. Shape optimization of Low Speed Airfoils using MATLAB and Automatic Differentiation