

Computational Fluid Dynamics I: Euler Equation Solver (FVM)

Gandharv Kashinath

University of Michigan, Ann Arbor, MI

11/14/2008

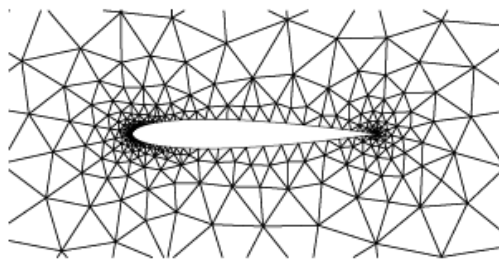
In this project a cell based finite volume method to solve the Euler equations of gas dynamics was implemented on unstructured, triangular meshes. Specifically, compressible flow around a NACA 0012 airfoil was modeled.

Introduction

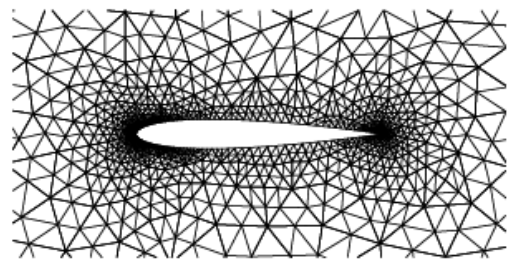
In this project a cell based finite volume method to solve the Euler equations of gas dynamics was implemented on unstructured, triangular meshes. Specifically, compressible flow around a NACA 0012 airfoil was modeled.

Problem Description

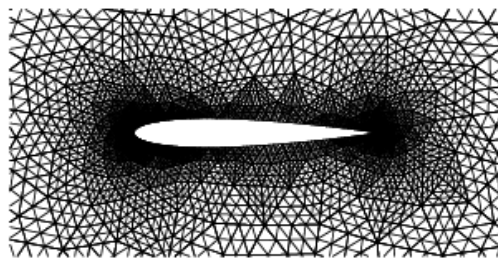
The emphasis of this project was on the implementation of; the numerical flux function and using it in a local-time stepping finite volume method for the 2-D Euler equations. The solution algorithm was then applied to NACA 0012 airfoil geometry for three different meshes (Figure 1), at various freestream Mach numbers, M_∞ , and angles of attack, α . The different angle of attack was simulated by adjusting the freestream flow angle. The airfoil geometry and mesh was fixed.



(a) Coarse mesh: 905 cells



(b) Medium mesh: 3620 cells



(c) Fine mesh: 14480 cells

Figure 1: Close-ups of the three different meshed used for the solution.

Farfield Boundary Conditions

The outer (i.e. farfield) boundary conditions was implemented through the flux function (i.e. eulerflux.m) using the state vector from the interior triangle as one of the state vectors and the freestream state vector as the other state vector.

Local Timestepping and CFL Definition

In order to accelerate convergence of the solution to steady state a local timestepping for each control volume was used. It was noted that the solution thus obtained was not accurate in time, but since the unsteady evolution of the flow was not of interest the lack of time accuracy was not a problem. To implement local timestepping, a vector of time steps was calculated, one timestep for each cell. Specifically the CFL number for a cell was defined to be,

$$CFL = \frac{\Delta t}{2A} \sum_{i=1}^3 |s|_i h_i \quad (1)$$

where A is the area of the cell, the summation is over the three edges of the cell, $|s|_i$ is the maximum propagation speed for the edge, and h_i is the length of the edge. For a cell j, a forward Euler integration of the finite-volume discretization of the conservation law was,

$$U_j^{n+1} = U_j^n - \left(\frac{\Delta t}{A} \right)_j R_j^n$$

where R_j^n is the standard finite volume flux residual for cell j:

$$R_j^n = \sum_{i=1}^3 \int_{edge\ i} \hat{F}(U_j^n, U_{N(j,i)}^n) \cdot n\ ds, \quad N(j,i) \text{ is the cell adjacent to } j \text{ across edge } i$$

Since the time stepping required the value of $\Delta t/A$ for each cell, the CFL number was used to calculate this by re-arranging Eqn. 1,

$$\frac{\Delta t}{A} = \frac{2CFL}{\sum_{i=1}^3 |s|_i h_i}$$

The summation of $|s|_i h_i$ was calculated during the flux calculations and a CFL value of 0.9 was used as the only restriction for stability was $CFL < 1$ in general.

Flux Calculation

The numerical flux on an edge between two cells as illustrated in Figure 2, was calculated in eulerflux.m. For interior edges, the HLLC (Harten, Lax, van Leer, Einfeldt) flux function to evaluate

$$\hat{H} \equiv \hat{F} \cdot \mathbf{n}:$$

$$\hat{H} \stackrel{\text{def}}{=} \hat{F} \cdot \mathbf{n} = \frac{1}{2}(H_L + H_R) - \frac{1}{2} \frac{s_{\max} + s_{\min}}{s_{\max} - s_{\min}} (H_L - H_R) + \frac{s_{\max} s_{\min}}{s_{\max} - s_{\min}} (U_L - U_R)$$

where $H_L = F(U_L) \cdot \mathbf{n}$ and $H_R = F(U_R) \cdot \mathbf{n}$, where $U = [\rho, \rho u, \rho v, \rho E]^T$, $H = F(U) \cdot \mathbf{n}$ is given by;

$$H = \begin{bmatrix} \rho u_n \\ \rho u u_n + p n_x \\ \rho v u_n + p n_y \\ \rho u_n E + u_n P \end{bmatrix}, \quad \mathbf{n} = [n_x, n_y], \quad u_n = u n_x + v n_y$$

The wave speeds were given by;

$$s_{\min} = \min(s_{L,\min}, s_{R,\min})$$

$$s_{\max} = \min(s_{L,\max}, s_{R,\max})$$

$$s_{L,\min} = \min(0, u_{nL} - c_L), \quad s_{L,\max} = \min(0, u_{nL} + c_L)$$

$$s_{R,\min} = \min(0, u_{nR} - c_R), \quad s_{R,\max} = \min(0, u_{nR} + c_R)$$

where c_L, c_R were the sound speeds and was given by;

$$c = \sqrt{\frac{\gamma P}{\rho}}$$

The maximum propagation speed, $|s| = \max(|u_{nL}| + c_L, |u_{nR}| + c_R)$, was computed while calculating the \hat{H} . As mentioned earlier for edges on the farfield boundary, eulerflux.m was used with the right state set to the freestream conditions.

For edges on the airfoil boundary, the wallflux.m was used in which the flux at a solid wall was computed. Since $u_n = 0$ at a solid wall, the boundary flux was due to the pressure:

$$H = \begin{bmatrix} 0 \\ p n_x \\ p n_y \\ 0 \end{bmatrix}$$

The maximum propagation speed at a boundary edge was, $|s| = \max(|u_n| + c)$, where c is the speed of sound for the boundary cell.

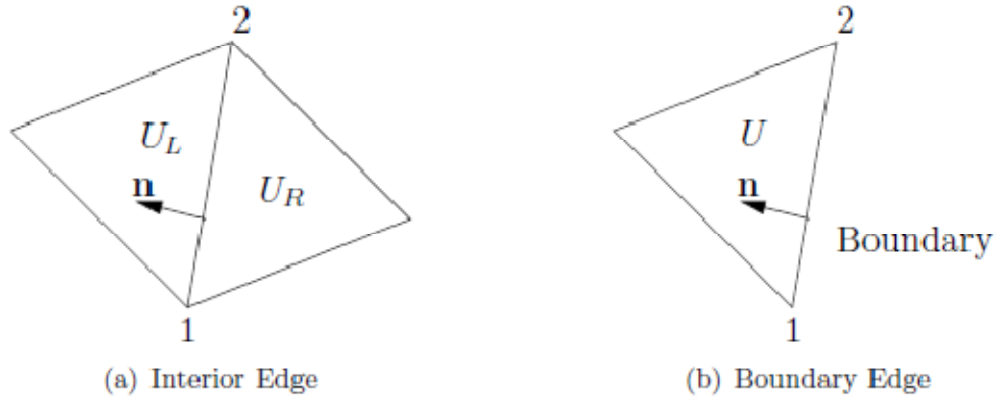


Figure 2: Definition of left and right states for edges. 1 and 2 indicate the endpoint nodes for the edge in question.

Geometry Calculations

In order to complete the flux and time-stepping calculations several geometric quantities of each control volume was needed. These included; the normal components at edge face, length of the each edge and the area of the control volume (here the equation is not presented, as during the formulation of the problem the area was never used explicitly). These quantities were calculated using basic analytical geometrical equations.

Consider an edge, whose end point coordinates are given to be (x_1, y_1) and (x_2, y_2) . In order to calculate the normal to the face a vector describing the edge was constructed from point 1 to point 2 given by;

$$\vec{v}_{12} = (x_2 - x_1)\hat{i} + (y_2 - y_1)\hat{j}$$

Now in order to obtain the normal to this edge the above vector was crossed with a unit vector pointing out of the plane as follows;

$$\hat{n} = \vec{v}_{12} \times \hat{k}$$

The normal vector was further normalized to obtain a unit normal and the following equation was obtained and used to calculate the normal needed for the flux calculations;

$$\hat{n} = \frac{-(y_2 - y_1)\hat{i} + (x_2 - x_1)\hat{j}}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}}$$

The length of the edge was calculated using the following equation;

$$s_{12} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Force Coefficients

The lift and drag coefficients on the airfoil were calculated by integrating the pressure force along the airfoil surface. The x and y forces was calculated as;

$$F_x = - \int_{airfoil} p n_x ds,$$

$$F_y = - \int_{airfoil} p n_y ds$$

These integrals were computed by summing over the boundary edges on the airfoil and calling wallflux.m to obtain pn_x and pn_y . The drag force was defined to be parallel to the freestream, while the lift force was perpendicular to it. Thus the drag and lift was computed as;

$$F_D = F_x \cos(\alpha) + F_y \sin(\alpha)$$

$$F_L = -F_x \sin(\alpha) + F_y \cos(\alpha)$$

where α , is the angle of attack. The lift and drag coefficients were computed as follows;

$$C_D = \frac{F_D}{M_\infty^2}, \quad C_L = \frac{F_L}{M_\infty^2}$$

Results and Discussion

The code was validated with the validation plots provided in the project statement and the following plots from the program were obtained;

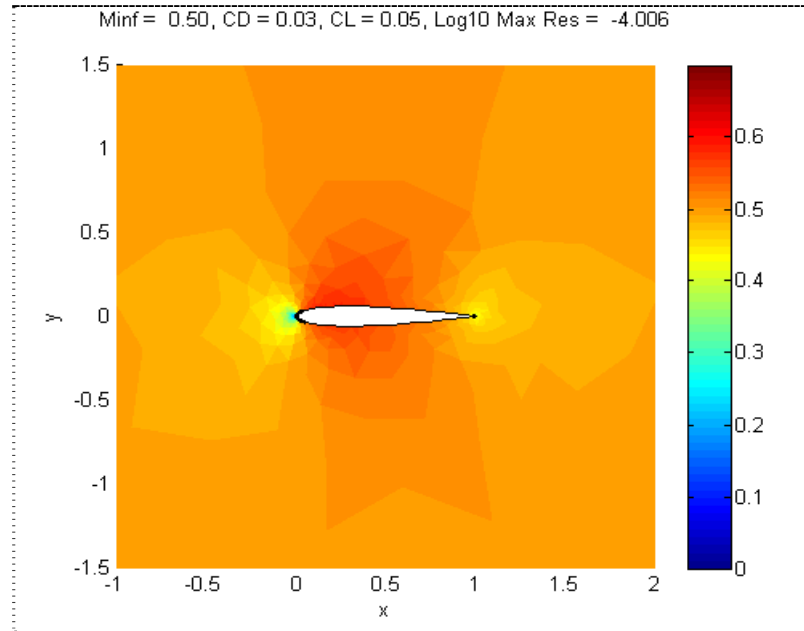


Figure 3: Coarse mesh: $M_\infty = 0.5$, $\alpha=1^\circ$, $C_L = 0.051$, $C_D = 0.031$.

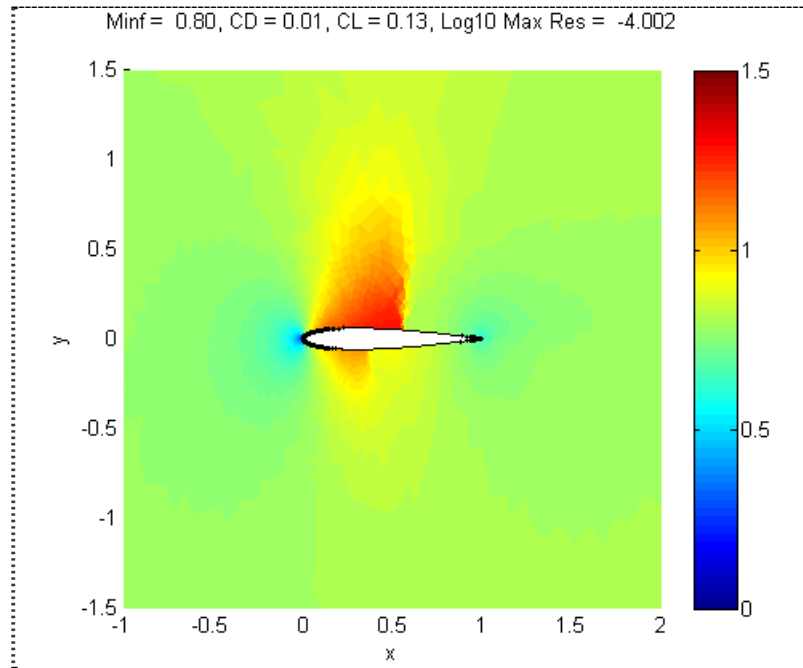
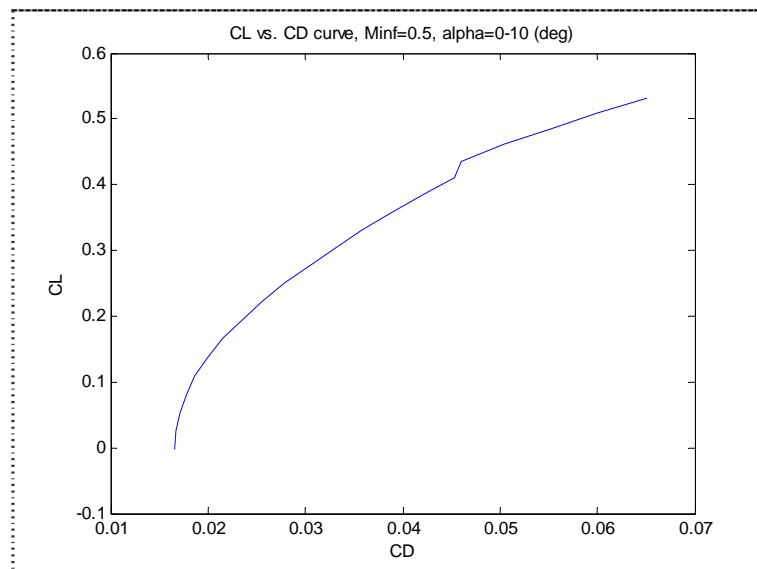


Figure 4: Fine mesh: $M_\infty = 0.8$, $\alpha = 1.25^\circ$, $CL = 0.13$, $CD = 0.01$.

Comparing the above plots with the validation plots presented in the project statement it was observed that the Mach number field match up exactly with one another. To further validate the code the coefficient of Lift, C_L and C_D were compared with the given values for the subsonic coarse mesh case which was found to be; $C_L = 0.051$ and $C_D = 0.031$ for a $M_\infty = 0.5$, $\alpha = 1^\circ$. These values are in good agreement with the values provided in the project statement.

Further using the medium mesh and a Mach number of 0.5, the code was run for $\alpha = [0, 10]$ with increments of 0.1 and the following plot for the C_L vs. C_D was obtained;



From the above plot it can be concluded that, with an increase in the angle of attack we get an increase in the C_L values, but at the same time as expected the C_D also increases. Thus, at higher angles of attack we definitely get a higher lift but at the same time we encounter an increase in drag. Observing the plot carefully it was found that at relatively lower angles of attack, the C_L increases at a higher rate than the C_D and at relatively higher angles of attack the C_L increases almost at the same rate as the C_D and this is exactly correlated with the physics of the problem.

From the plot we also observe a small kink in the values of C_L and C_D and this can be attributed to numerical errors during the simulation. This kink can be resolved by converging the solution to a lower residual value or by refining the mesh for that particular angle of attack.

Further the code was run for the two “classic” test cases for the Euler equations using the medium and fine meshes and the results for those two cases are as follows;

Case 1: $M_\infty = 0.8$, $\alpha = 1.25^\circ$:

a) Medium mesh:

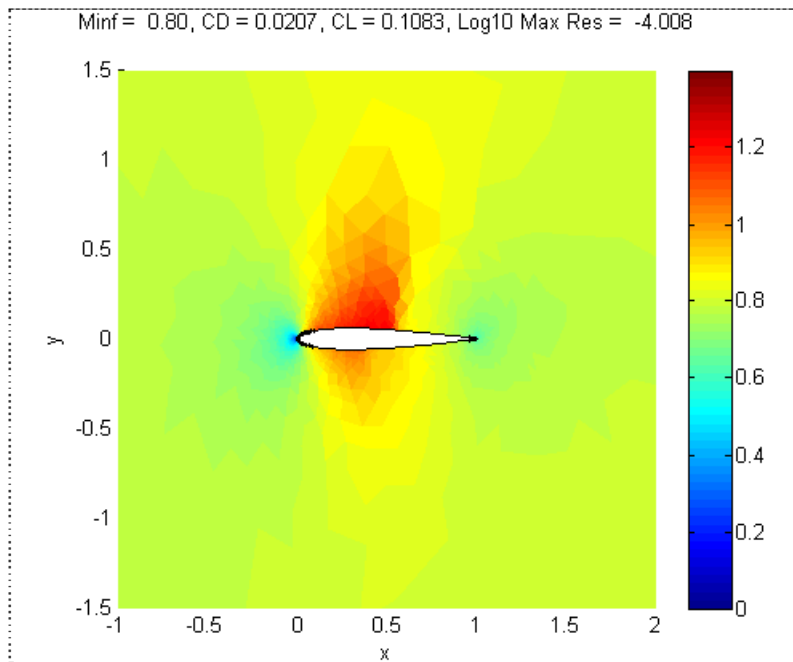


Figure 5: Medium mesh, $M_\infty = 0.8$, $\alpha = 1.25^\circ$, $C_L = 0.1083$, $C_D = 0.0207$.

b) Fine mesh:

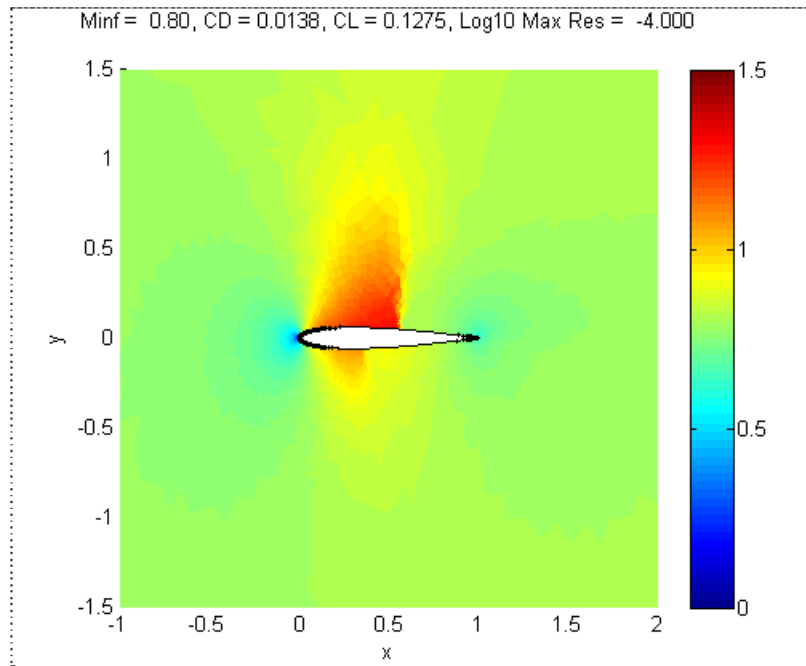


Figure 6: Fine mesh, $M_\infty = 0.8$, $\alpha = 1.25^\circ$, $C_L = 0.1275$, $C_D = 0.0138$.

From the above two plots it can be seen that the formation of the shock occurs on the upper surface of the airfoil (almost a normal shock closer to the airfoil surface) where there is a sudden jump in Mach number from supersonic to subsonic speeds. The location is roughly between $x = 0.65$ - 0.75 . The formation of this shock was on the upper surface as expected due to a positive angle of attack. This shock formation was more pronounced in the fine mesh solution compared to the medium mesh.

The following values for C_L and C_D were obtained from this case;

Medium mesh: $C_L = 0.1083$, $C_D = 0.0207$.

Fine Mesh: $C_L = 0.1275$, $C_D = 0.0138$.

Case 2: $M_\infty = 0.95$, $\alpha = 0^\circ$:

a) Medium Mesh:

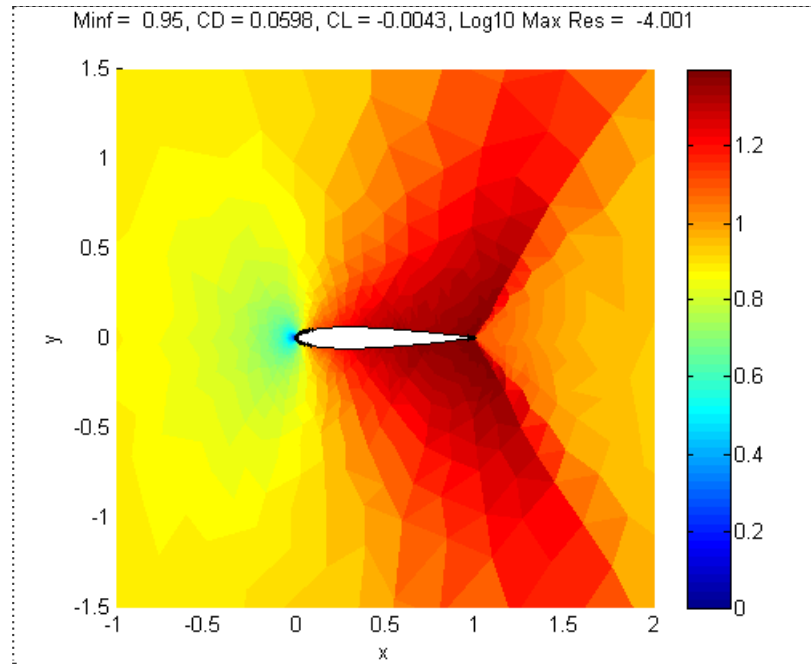


Figure 7: Medium mesh, $M_\infty = 0.95$, $\alpha = 0^\circ$, CL = -0.0043, CD = 0.0598.

b) Fine Mesh:

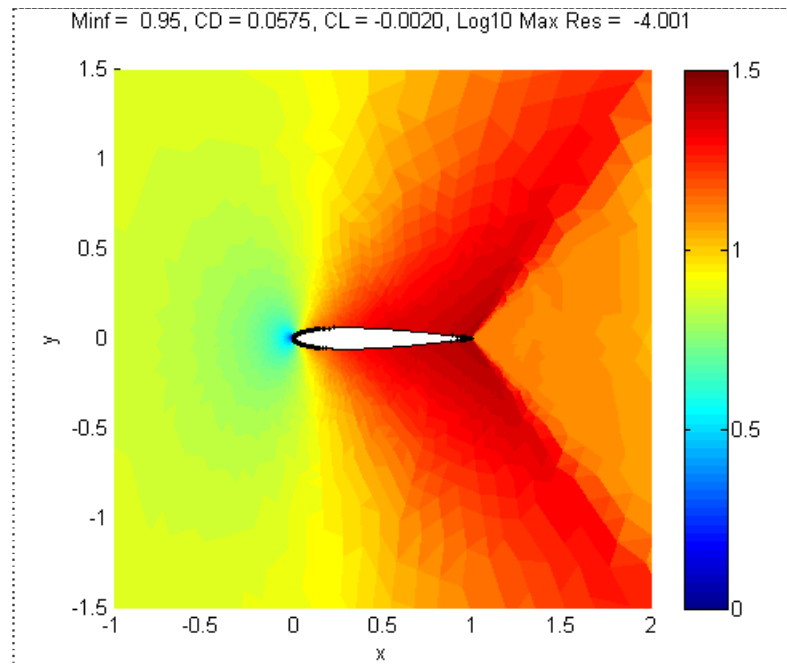


Figure 8: Fine mesh, $M_\infty = 0.95$, $\alpha = 0^\circ$, CL = -0.0020, CD = 0.0575.

From the above two plots it can be seen that the formation of the shock was closer to the trailing edge (or at the trailing edge) of the airfoil. Since the flow was symmetric (i.e. angle of attack of 0) the shocks were present both on the upper and lower surfaces. Since the Mach number reduces from a high supersonic speed to a low supersonic speed (close to 1.0) it can be concluded that an oblique shock was formed, both on the upper and lower surfaces. This is further confirmed by the Mach number contours at the trailing edge in both the medium and fine mesh solutions. The shock formation in either of the two meshes was comparable, with a little more resolution seen in the fine mesh contour. Note the shock formation for this initial flow conditions was not as pronounced as the previous case.

The following values for C_L and C_D were obtained from this case;

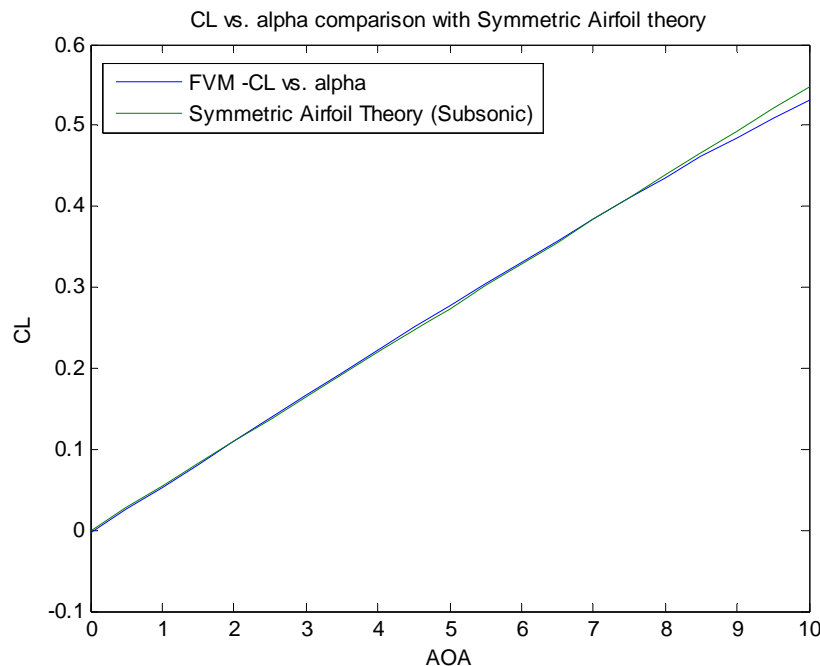
Medium mesh: $C_L = -0.0043$, $C_D = 0.0598$.

Fine Mesh: $C_L = -0.0020$, $C_D = 0.0575$.

From the above two values a low negative lift value is obtained for zero angle of attack. This is in contradiction to the symmetric airfoil theory and experimental results, which states that $C_L=0$ for zero angle of attack, but this can be attributed to the fact that the mesh considered for the solution was not exactly symmetric, which meant that the flow simulated over the airfoil was not exactly symmetric. Since the value of this negative lift is very small it can be concluded that the mesh is good enough for other calculations. However, this error can be overcome by considering a perfectly symmetrical mesh.

Additional Results

An attempt was made to compare the results obtained from the subsonic i.e. $M_\infty=0.5$ with the classical thin airfoil theory (symmetric case) which states that the slope of the C_L vs. α plot should be equal to 2π . In order to do this the following plot of C_L vs. α was constructed;



where the symmetric airfoil theory was generated by using;

$$C_L = 2\pi\alpha$$

From the above plot we clearly see that the two slopes match up very well and the numerical result is in good agreement with the theoretical prediction.

Appendix

Pressure required during the formulation was derived and the following equation was used;

$$P = (\gamma - 1) \left(\rho E - \left(\frac{1}{2} \rho (u^2 + v^2) \right) \right)$$

Comments on MATLAB files:

Several Matlab-related files are available in a zip file. The following is a brief description of these files:

- **FVM.m**: This is the main script which you should modify to implement the finite volume method. Comments inside the script indicate where additions need to be made. Note, even if you do not choose to use Matlab, this file is still useful for understanding the mesh structure and the layout of the solution algorithm. A restart capability exists so that additional iterations can be made from a solution that is optionally passed in. This is useful to further converge a solution on a current mesh.
- **eulerflux.m**: This function calculates the two-dimensional upwinded flux given the left and right state vectors, a unit normal vector pointing from the right triangle to the left triangle, and the ratio of specific heats. It returns the flux entering the left triangle and the maximum magnitude propagation speed on the edge (this propagation speed is used in calculating the timestep). Note: the flux entering the right triangle is just the opposite of the flux entering the left triangle. This is where you will implement the HLLE flux function.
- **wallflux.m**: This function calculates the two-dimensional flux at a wall boundary edge given the interior state vector, the unit normal vector pointing into the computational domain, and the ratio of specific heat. It returns the flux entering the computation domain and the maximum magnitude propagation speed on the edge (this propagation speed is used in calculating the timestep). You need to implement the pressure-only numerical flux here.
- **Mesh_coarse.mat**, **Mesh_medium.mat**, **Mesh_fine.mat**: These are Matlab binary mesh files containing the structures documented in **FVM.m**. To load a mesh into the Matlab workspace, type “load mesh_coarse.mat” (for example). If you are not using Matlab for the main code, you will need to initially use Matlab to load these meshes and convert them to the output format that you need. See the documentation in **FVM.m** on the mesh structure.

The above description was obtained from the project description document provided by the instructor.