2D Euler Equation Solver For Transonic Flows

Muhammad Baqui

December 16, 2013

1 Introduction

Euler equation is solved for flow over NACA0012 airfoil in 2D domain. The domain is discritized with unstructured triangular grid. A cell centered finite volume scheme is used for spatial discritization. The temporal discritization is done with low storage Runge Kutta scheme. The computer program is impelemented with C++ programming language. The Euler equation for 2D can be presented as :

$$(\boldsymbol{u})_{,t} + \nabla \boldsymbol{F} = 0 \tag{1}$$

For cell centered finite volume the equation results in:

$$\frac{\partial u}{\partial t} = -RHS$$
$$RHS = \Sigma_{faces} \quad \boldsymbol{F}.\boldsymbol{n}$$

The RHS is evaluated with the following flux contributions.

$$\begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho e \end{bmatrix} = \begin{bmatrix} \rho u & \rho v \\ \rho u^2 + p & \rho uv \\ \rho uv & \rho v^2 + p \\ (\rho e + p)u & (\rho e + p)v \end{bmatrix}$$

The unknows for Euler equations are presented as the left column of the above equation and flux components are shown as the right column . The normals are evaluated as

$$\boldsymbol{n} = (n_x, n_y) = (\Delta y, -\Delta x)$$

2 Solution Approach

The problem is solved in C++ programming language. Structure feature of C++ language is used to store cell level unknowns,fluxes and other physical properties of a cell. Along with cell data structure, element connectivity, coordinate information and boundary nodes are required. 1D Arrays are used to store these elements. To calculate fluxes through each cell, edges/faces of that cell, neighbour cells surrounding a face information is required. These informations are obtained following the algorithms presented in the class by Dr. Rainald Lohner. Element surrounding element information is also needed and

it is implemented following a brute force approach. A typical cell data strucure can be seen below:
struct Cell {
/* geometric properties */
cell id,area,normals,units normals,boundary type,
face nodes,time step,neighbours;
/* flow properties */
pressure,density,densityVelocity,densityEnergy
/* fluxes */
fluxDensity,fluxDensityVelocityu,fluxDensityVelocityv,
fluxDensityEnergy;
/* previous timestep values of unkowns */
oldDensity,oldDensityVelocity,oldDensityEnergy
}

The neighbours of a perticular cell is stored in Vector<> data structure of C++. A nice thing about vectors are one does not need to specify the size of the structure. As neighbours surrounding a perticular cell can vary, so vector is found to be suitable choice for this case. The solution process can be summerized as follows

1. Read Mesh file
2. formulate coordiante array
3. formulate connectivity array
4. formulate boundary elements
5. find elements surrounding elements
6. calculate freestream/farfield conditions
7. put initial conditions in all cells
8. enter time loop
9. loop through faces of elements
    - loop through faces of elements
    - calculate fluxes
    - obtain artificial viscosity
    - update fluxes
    - update unknowns
10. update boundary values
11. print results

The functions used for these purposes are shown below:
double Areacalc(int a,int b, int c);
double Centroidcalcx(int a,int b,int c);
double Centroidcalcy(int a,int b,int c);
int FindNeighbor(int ielemNode,int nelem);

int ComputeNormal(int ielemNode,int nelem);
double ComputeDensityFlux(int ielem,int nelem);
double ComputeDensityVelocityuFlux(int ielem,int nelem);
double ComputeDensityVelocityvFlux(int ielem,int nelem);
double ComputeDensityEnergyFlux(int ielem,int nelem);
double ArtificialViscosityDensity(int ielem,int nelem);
double ArtificialViscosityDensityVelocityu(int ielem,int nelem);
double ArtificialViscosityDensityVelocityv(int ielem,int nelem);
double ArtificialViscosityDensityEnergy(int ielem,int nelem);
int CorrectBoundaryValues(int nelem);
int calcInwardNormal(int ielem,int a1,int b1);
int ReadMeshFile(int npoin,int nelem,int bcnodes);
double convertToDouble(std::string const s);
int CalcBoundaryFaces(int npoin,int nelem,int bcnodes);
int WriteDatatoFile(int npoin,int nelem);

Evaluation of fluxes: There are four flux components: density,densityVelocityu,densityVelocityv,density These fluxes are evaluated at each face of a cell using an average of the two cell unknowns. For exam for edge a-b between two elements, the flux will be

$$F = \frac{1}{2}(F^a + F^b) \tag{2}$$

Fluxes at Boundary cells: Special treatment is needed for evaluation of fluxes through the boundary cells. There are two special boundary elements : (a) Farfield Boundary and (b) Wall Boundary. For farfield boundary farfield/initial conditions are considered in evaluation of fluxes. If a cell is at the farfield boundary its flux is evaulated as

$$F = \frac{1}{2}(F_{cell} + F_{fafield}) \tag{3}$$

In case of wall boundary only flux computed is the densityVelocityu and densityVelocityv fluxes. In these cases only pressure in added in the flux component as found in the class lecture slides (simple extrapolation of pw to p1).

Time Step Calculation : Time steps are calculated at each cell using the following formula:

$$\Delta t_{cell} = \frac{1}{2}\frac{\sqrt{A_{cell}}}{V_{cell} + (Speedofsound_{cell})} \tag{4}$$

3 Results

Figure 1 shows the pressure profile of the entire flow field. Shock waves can be seen red color along the leading edge of the airfoil. Figure 2 shows the pressure,density,energy and mach number distribution along the chord of the airfoil. Figure 3 shows the pressure profile of the entire flow field. Shock waves can be seen red color along the leading edge of the airfoil. Figure 4 shows the pressure,density,energy and mach number distribution along the chord of the airfoil.

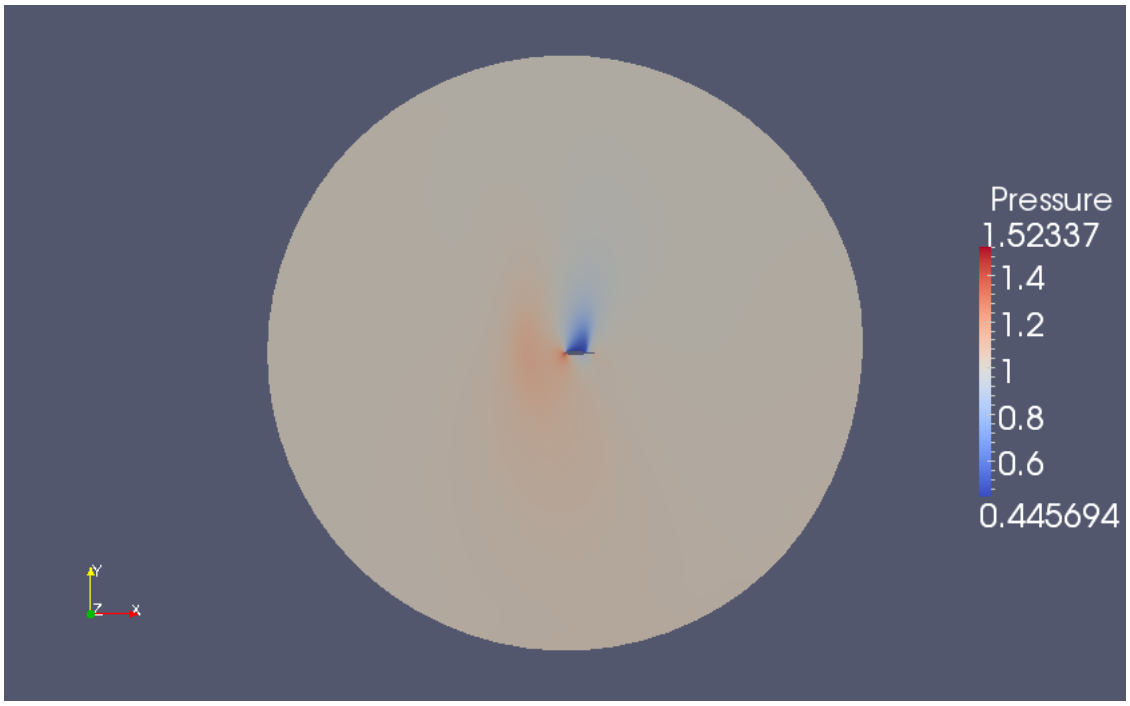
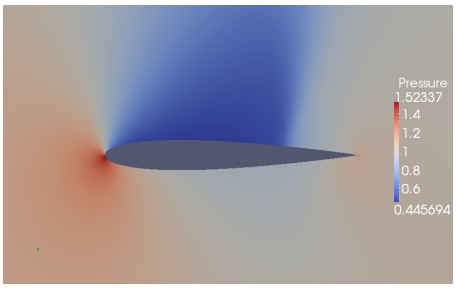
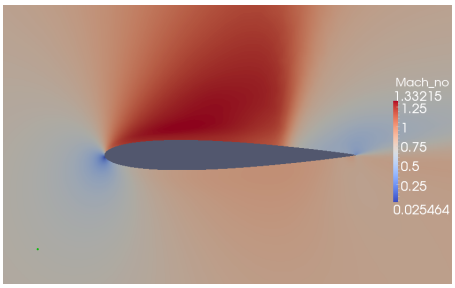


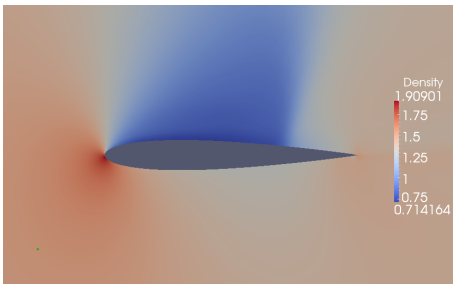
Figure 1: Pressure profile at transonic flow Ma 0.8 alpha 8 deg



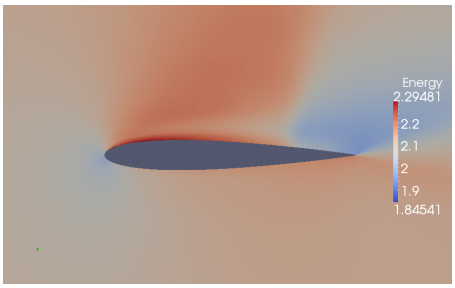
Pressure Distribution



Mach number



Density Distribution



Energy Distribution

Figure 2: Results for 8 degree angle of attach at Ma 0.8

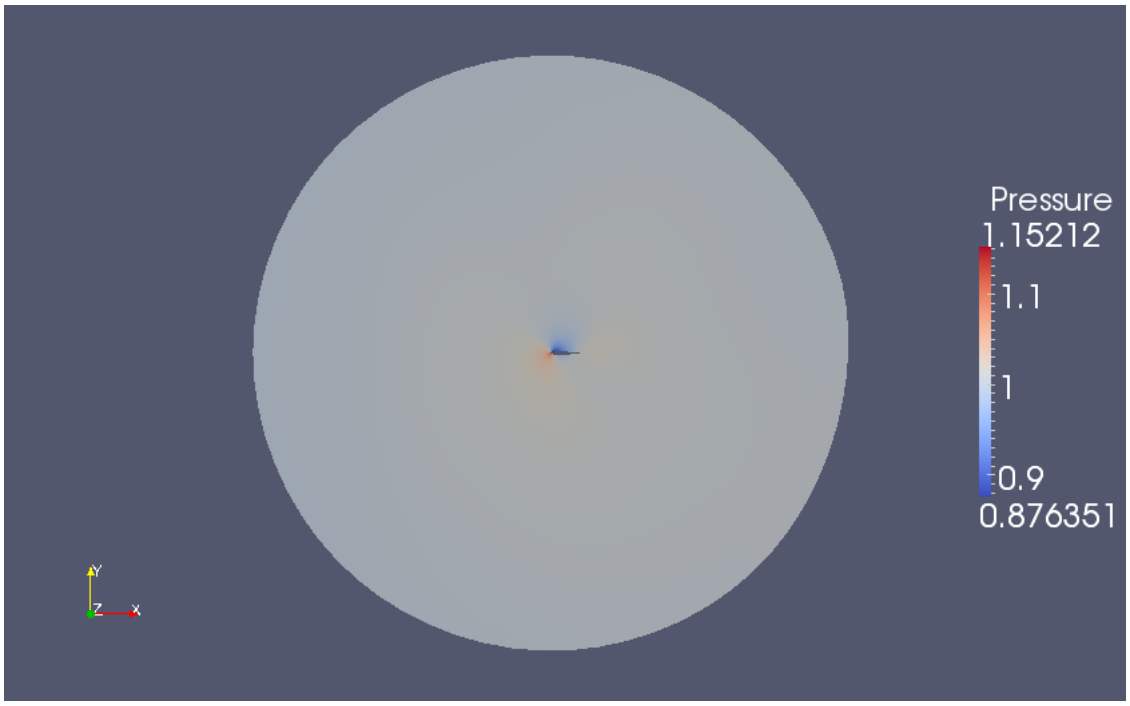
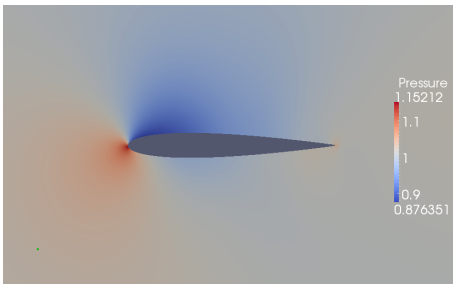
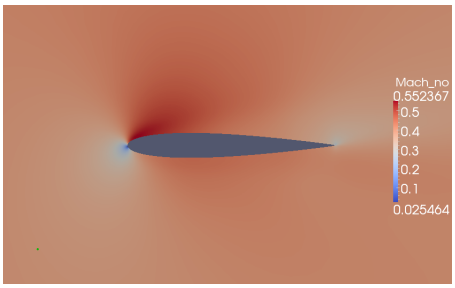


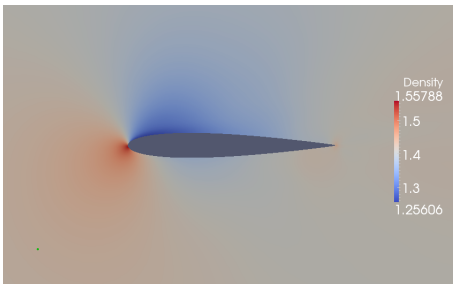
Figure 3: Pressure profile at transonic flow Ma 0.4 alpha 4 deg



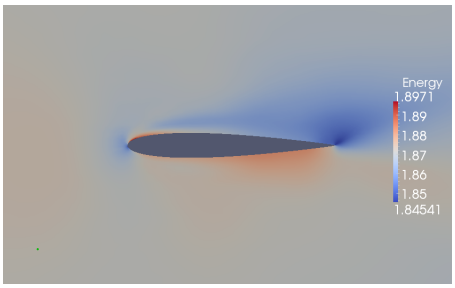
Pressure Distribution



Mach number



Density Distribution



Energy Distribution

Figure 4: Results for 4 degree angle of attach at Ma 0.4