Project 2 Report

Name: CHEN Yufan

Date: 2024/11/25

1 Introduction of the Problem (2-D blunt-body problem)

1.1 Problem Description and Initial Conditions

2-D supersonic flow over a 2-D cylinder with a radius of 20 mm at zero angle of attack. The computational grid has 160 (circumferential) × 80 (wall normal) cells in PLOT3D format. Numerically solve the problem using the finite-volume (FV) method. (The flow conditions: M_∞=8.1, T_∞=63.73 K, p_∞=370.6 Pa; the boundary conditions: (1) Upper boundary is the supersonic inflow (2) Lower boundary is set as inviscid wall (3) Left & right boundaries are set to be supersonic outflow; The grid data is given by PLOT3D format.)

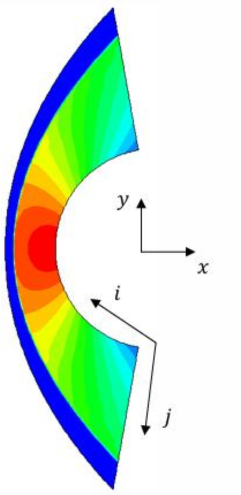


Figure 1 Description of 2-D blunt-body problem (numerical solution example)

1.2 Requirements

(1) Use second-order reconstruction with different limiters.

(2) Use different Riemann solvers (at least include HLL and AUSM).

(3) Use first-order difference formula for time discretization (monitor the change in density to check convergence).

(4) Choice for time step Δt= 1×10−8 s would be a good guess corresponding to a CFL number of approximately 0.2.

(5) Compare the numerical solution with the empirical shock shape (see Sec. 5.4 in Hypersonic and High-Temperature Gas Dynamics, J. D. Anderson Jr.).

(6) Compare the numerical surface pressure distribution with Newtonian theory.

(7) Submit your report (describe the detailed procedure and attach the source codes).

2 Numerical Solution of the Problem

For the 2-D blunt-body problem described in Chapter 2, the finite volume (FV) numerical solution **MATLAB codes** of 2D Euler Equations was obtained using **HLL & AUSM Riemann solver.** The solution vectors were then reconstructed using 2nd order schemes based on the **Min-Mod limiters** for Riemann solver. What's more, the differences between the numerical solution results and the Newton theoretical solutions are compared, also the empirical shock shape is compared with numerical one.

2.1 Settings and Procedures of Finite Volume Method

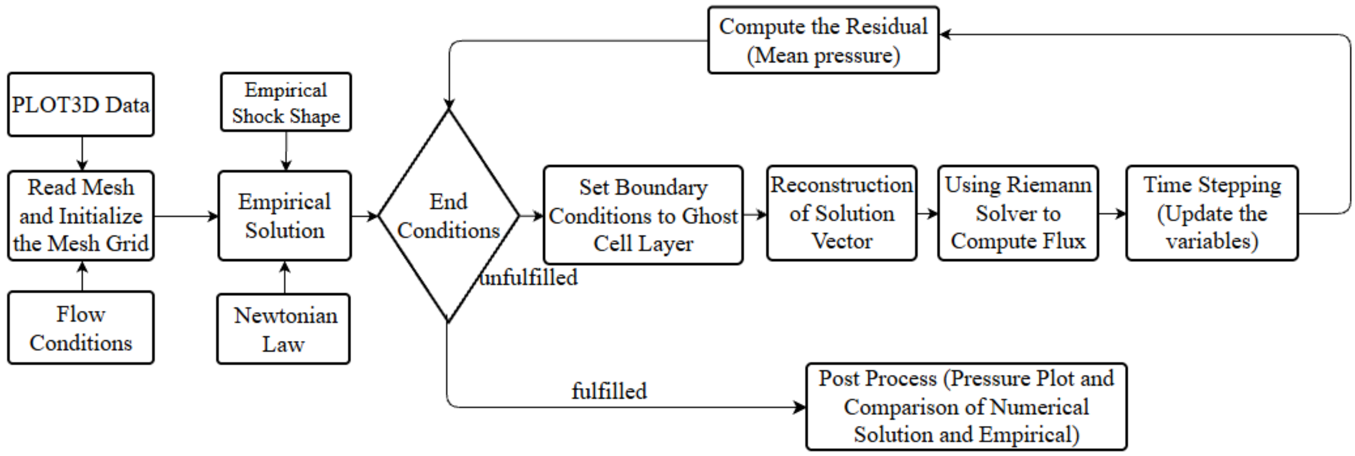

Compute the Residual (Mean pressure)
PLOT3D Data
Empirical Shock Shape
Read Mesh and Initialize the Mesh Grid
Empirical Solution
End Conditions
Set Boundary Conditions to Ghost Cell Layer
Reconstruction of Solution Vector
Using Riemann Solver to Compute Flux
Time Stepping (Update the variables)
unfulfilled
Flow Conditions
Newtonian Law
fulfilled
Post Process (Pressure Plot and Comparison of Numerical Solution and Empirical)

Figure 2 The procedures of FVM solver

FVM discretization: Before Setting the computational domain, the FVM discretization method needs to be given. Regarding computing the 2D inviscid flow, the conservative form of Euler equations is written as blown.

$$\frac{\partial U}{\partial t}+\frac{\partial F}{\partial x}+\frac{\partial G}{\partial y}=0 \tag{1}$$

$$U=\begin{pmatrix}\rho\\ \rho u\\ \rho v\\ \rho E\end{pmatrix}, F=\begin{pmatrix}\rho u\\ \rho u^2+p\\ \rho uv\\ \rho Eu+pu\end{pmatrix}, G=\begin{pmatrix}\rho v\\ \rho uv\\ \rho v^2+p\\ \rho Ev+pv\end{pmatrix}, E=\frac{1}{2}(u^2+v^2)+\frac{p/\rho}{\gamma-1}$$

By using FVM method, the equation (1) is supposed to be written into integral form as equation (2) shown blown:

$$\frac{\partial}{\partial t}\int Ud\Omega+\oint\vec{F}\cdot\vec{n}dS=0 \tag{2}$$

$$\vec{F}=F\vec{\imath}+G\vec{\jmath} \tag{3}$$

$$\vec{F}\cdot\vec{n}=F_n=\begin{pmatrix}\rho u_n\\ \rho uu_n+pn_x\\ \rho vu_n+pn_y\\ \rho Eu_n+pu_n\end{pmatrix}, u_n=un_x+vn_y \tag{4}$$

Then in the structured grid, the equation (2) can be written as equation (5) for space discretization:

$$\frac{dU_{I,J}}{dt}+\frac{1}{\Omega_{I,J}}\sum_{faces}^{n}F_n\Delta S=0 \tag{6}$$

Settings:

(1) Meshing settings: the data of mesh is given by PLOT3D format, which can read and visualize by the ***ReadPLOT3D*** function in the codes, which gives a visualization of the mesh that given by the mesh data. And the normal vector of each cell wall, the length of each wall of cells, the area of each cell and the position of each center point of cell are computed by ***MeshSetting*** function in the codes.

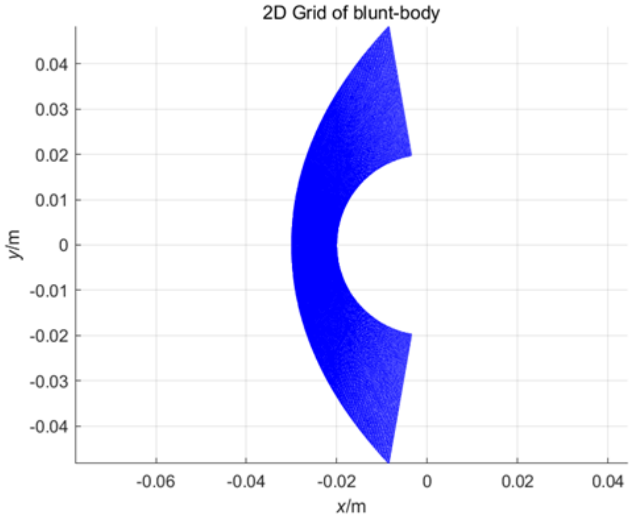


Figure 3 Mesh Schematic

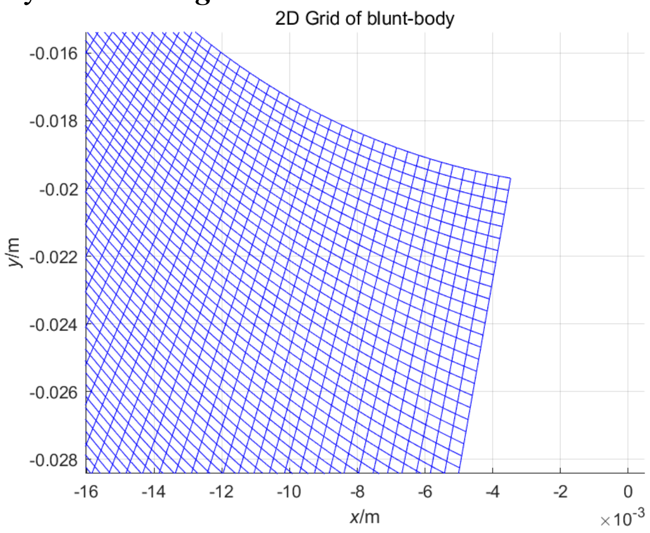


Figure 4 Mesh Schematic (local zoom)

(2) Initial conditions: use M_∞=8.1, T_∞=63.73 K, p_∞=370.6 Pa to initialize the 160x80 cells' physical variables. And the Δt is set to be 1×10^{-8} s.

(3) Boundary conditions: Due to the reconstruction scheme is 2nd order scheme, there are 2 layers of ghost cells outside of the real cells. (***Boundary*** function in the codes)

The left & right boundaries are set to be supersonic outflow, which means that the physical variables of two layers of ghost cells are supposed to be set as the same value as the last layer of real cells.

The upper boundary is the supersonic inflow, which means that the physical variables of two layers of ghost cells are supposed to be set as the same value as the same value of flow conditions.

The lower boundary is set as the inviscid wall, which means that the wall should be equal to the symmetric boundary, the normal velocity at the boundary (the wall of cylinder) is zero, the velocity are supposed to be set as Figure 4, and other physical variables of ghost cells layer are supposed to be symmetric to the real cells layer.

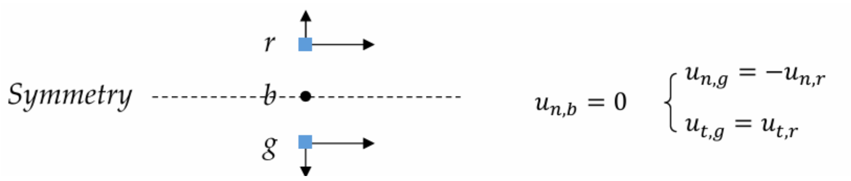


Figure 5 The velocity settings in the inviscid wall

(4) End conditions: the end conditions can be controlled by the maximum timestep, running time and pressure residuals. For steady problem in this case, the pressure residual is selected to control the ending.

Procedure in the Main Loop:

(1) Reconstruction of solution vector: the solution vectors are reconstructed by nonlinear second order monotonic scheme with MinMod limiter. (***Reconstruction*** function in the codes)

$$\begin{cases} U_L = U_I + \frac{1}{2}\varphi(r_I)(U_I - U_{I-1}) \\ U_R = U_{I+1} - \frac{1}{2}\varphi\left(\frac{1}{r_{I+1}}\right)(U_{I+2} - U_{I+1}) \end{cases}, r_I = \frac{U_{I+1} - U_I}{U_I - U_{I-1}} \tag{7}$$

(2) Compute flux by Riemann solver: the key in the FVM method is compute the flux around each grid cell, in this case the grid is the structured quadrilateral, $F_{I+\frac{1}{2},J}, F_{I-\frac{1}{2},J}, F_{I,J+\frac{1}{2}}, F_{I,J-\frac{1}{2}}$ are supposed to be solve in the each iteration. There are two type of Riemann solvers using in the codes:

HLL Scheme:

$$F_{I+\frac{1}{2}} = \begin{cases} F_L \quad if\ 0 \le S_L \\ \dfrac{S_R F_L - S_L F_R + S_L S_R (U_R - U_L)}{S_L - S_R} \quad if\ \ S_L \le 0 \le S_R \\ F_R \quad if\ S_R \le 0 \end{cases} \tag{8}$$

HLL scheme as equation (8) is for the 1-D case, for 2-D problem, the local coordinate rotation needs to be implemented, which can be illustrated in Figure 6, u' is the normal velocity and v' is tangential velocity which needs to be deemed as **passive variable** when using HLL scheme. (***HLL*** function in the codes)

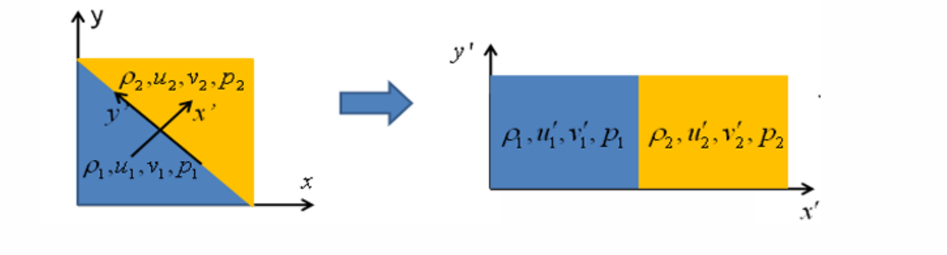


Figure 6 Transform into a 1-D problem by rotation of axes

AUSM Scheme:

$$F_n = \begin{pmatrix} \rho u_n \\ \rho u u_n + p n_x \\ \rho v u_n + p n_y \\ \rho H u_n \end{pmatrix} = \rho {M_n}^{\pm} a \begin{pmatrix} 1 \\ u \\ v \\ H \end{pmatrix} + \begin{pmatrix} 0 \\ p^{\pm} n_x \\ p^{\pm} n_y \\ 0 \end{pmatrix} \equiv F_c + F_p, M_n = \frac{u_n}{a} \tag{9}$$

$${M_n}^{+}(U_L) = \begin{cases} M_n & if M_n > 1 \\ \dfrac{(M_n+1)^2}{4} & if |M_n| \le 1 \\ 0 & if M_n < -1 \end{cases}, \quad {M_n}^{-}(U_L) = \begin{cases} 0 & if M_n > 1 \\ -\dfrac{(M_n-1)^2}{4} & if |M_n| \le 1 \\ M_n & if M_n < -1 \end{cases} \tag{10}$$

$$p^{+}(U_L) = \begin{cases} p & if M_n > 1 \\ \dfrac{(M_n+1)^2}{4}(2 - M_n) & if |M_n| \le 1 \\ 0 & if M_n < -1 \end{cases}, \quad p^{-}(U_L) = \begin{cases} 0 & if M_n > 1 \\ -\dfrac{(M_n-1)^2}{4}(2 + M_n) & if |M_n| \le 1 \\ p & if M_n < -1 \end{cases} \tag{11}$$

The AUSM scheme as equation (9) splits the Mach number term and pressure term into negative and positive term respectively which can replace by left and right primitive variables given by reconstruction. What' more, the tangential velocity in this scheme is supposed to be deemed as passive term. (***AUSMplus*** function in the codes)

(3) Time marching: this code uses the first order time stepping scheme, which can be described as equation

(12). Due to this code compute 4 flux around each by each normal vector, the sum of flux is supposed to be purely add.

$$U_{I,J}^{n+1} = U_{I,J}^{n} + F_{I+\frac{1}{2},J} S_{I(i,j)} + F_{I-\frac{1}{2},J} S_{I(i,j)} + F_{I,J+\frac{1}{2}} S_{J(i,j+1)} + F_{I,J-\frac{1}{2}} S_{J(i,j)} \tag{12}$$

3 Results

3.1 Newton theory

Using the Newton theory, we can get the pressure at center point of the cylinder wall, which is maximum pressure in the computational domain.

$$C_{p_{max}} = \frac{2}{\gamma M_\infty^2}\left\{\left[\frac{(\gamma+1)^2 M_\infty^2}{4\gamma M_\infty^2 - 2(\gamma-1)}\right]^{\frac{\gamma}{\gamma-1}}\left[\frac{1-\gamma+2\gamma M_\infty^2}{\gamma+1}\right] - 1\right\} \tag{13}$$

$$C_p = C_{p_{max}} \sin\theta^2 \tag{14}$$

By using the ***NewtonianLaw*** function in the code, we can get the pressure distribution at cylinder wall and the maximum pressure is about **31478 Pa**.

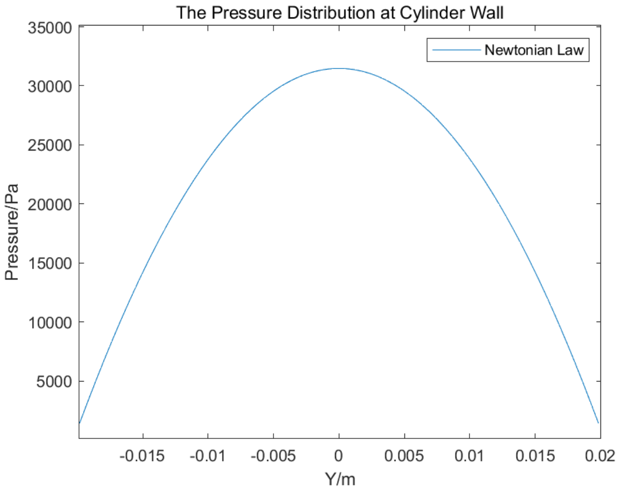
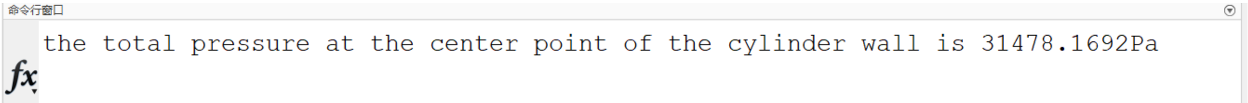


Figure 7 Newtonian Law pressure distribution



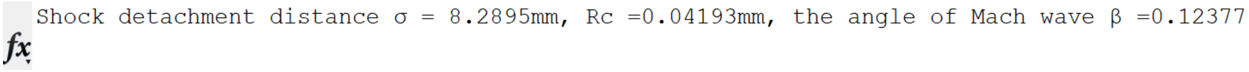
```
the total pressure at the center point of the cylinder wall is 31478.1692Pa
```

3.2 Empirical shock shape

The correlations given by the experiments results of Billig is used to predict the empirical shock shape (σ is the shock detachment distance, β is a Mach wave angle, R_c is the radius of curvature of the shock wave at the vertex of the hyperbola):

$$x = R + \sigma - R_c \cot \beta^2 \left[\left(1 + \frac{y^2 \tan \beta^2}{R_c^2} \right)^{\frac{1}{2}} - 1 \right], where \begin{cases} R_c = 1.386R \exp\left(\frac{1.8}{(M_\infty^2 - 1)^{0.75}} \right) \\ \sigma = 0.386R \exp\left(\frac{4.76}{M_\infty^2} \right) \\ \beta = \sin^{-1} \frac{1}{M_\infty} \end{cases} \qquad (15)$$

By using the function ***EmpiricalShockShape*** in the code, we can get the empirical shock shape of this case as shown in Figure 5.



```
Shock detachment distance σ = 8.2895mm, Rc =0.04193mm, the angle of Mach wave β =0.12377
```

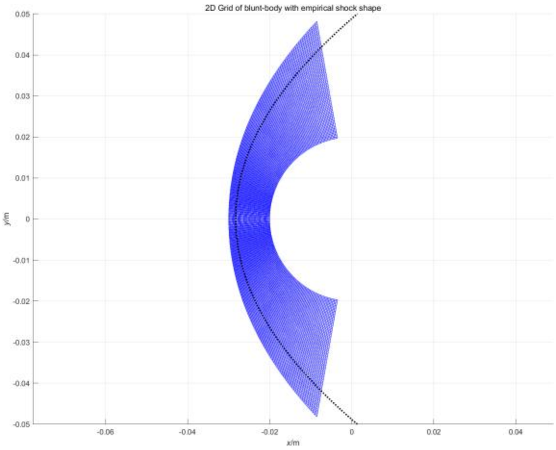


Figure 8 Empirical shock shape

3.3 Numeraical Solutions

(1) HLL Scheme:

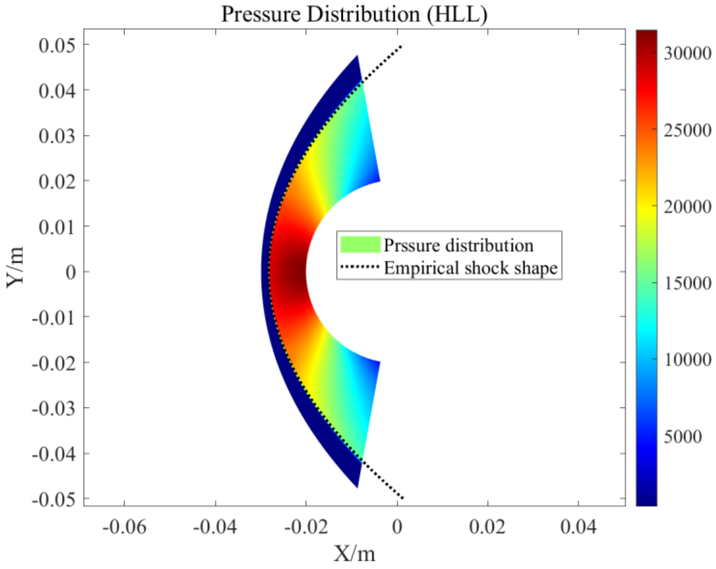


Figure 9 The pressure distribution at steady state (HLL)

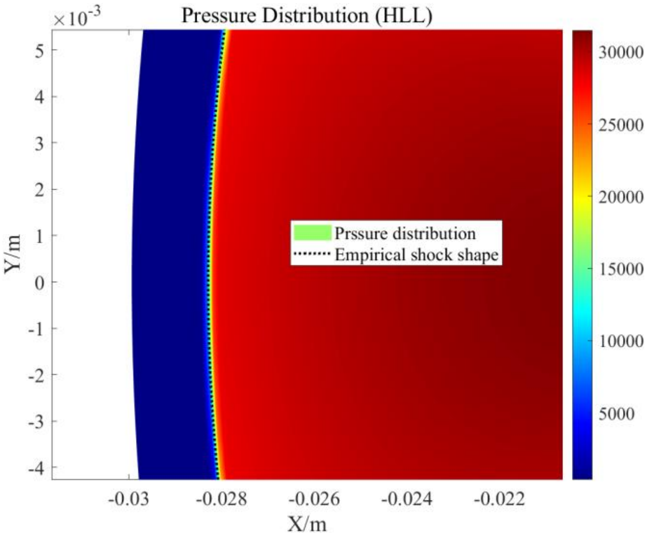
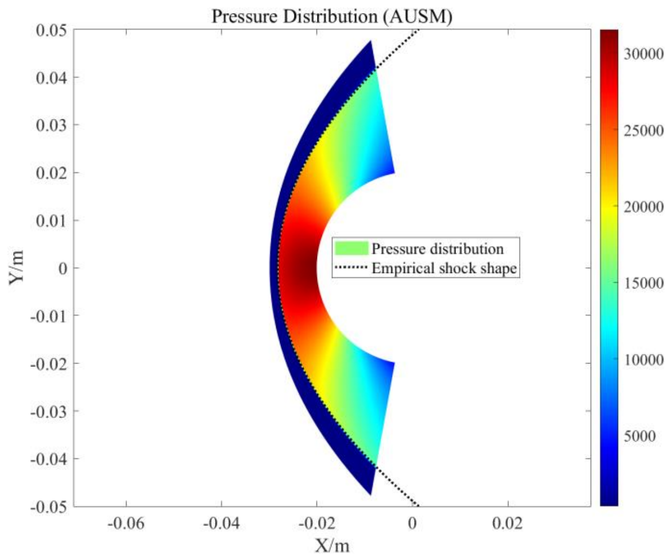


Figure 10 The pressure distribution at steady state (local zoom) (HLL)

(2) AUSM Scheme:



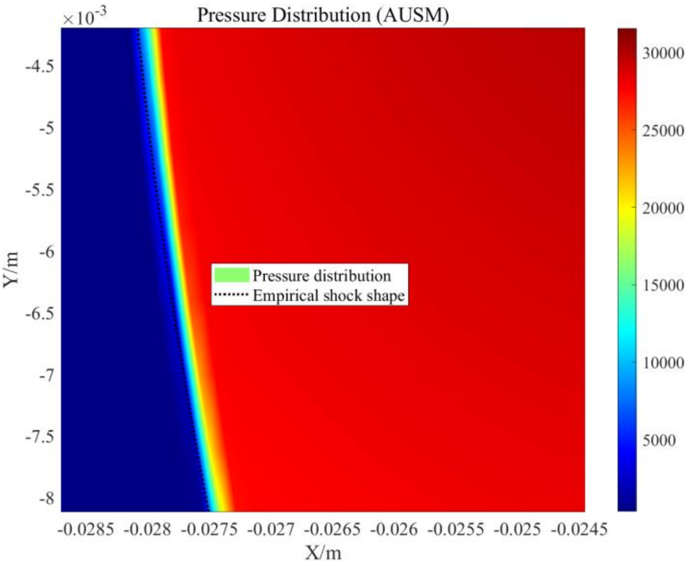


Figure 11The pressure distribution at steady state (AUSM)

Figure 12 The pressure distribution at steady state (local zoom) (AUSM)

(3) The Comparison of Empirical Method and Numerical Method:

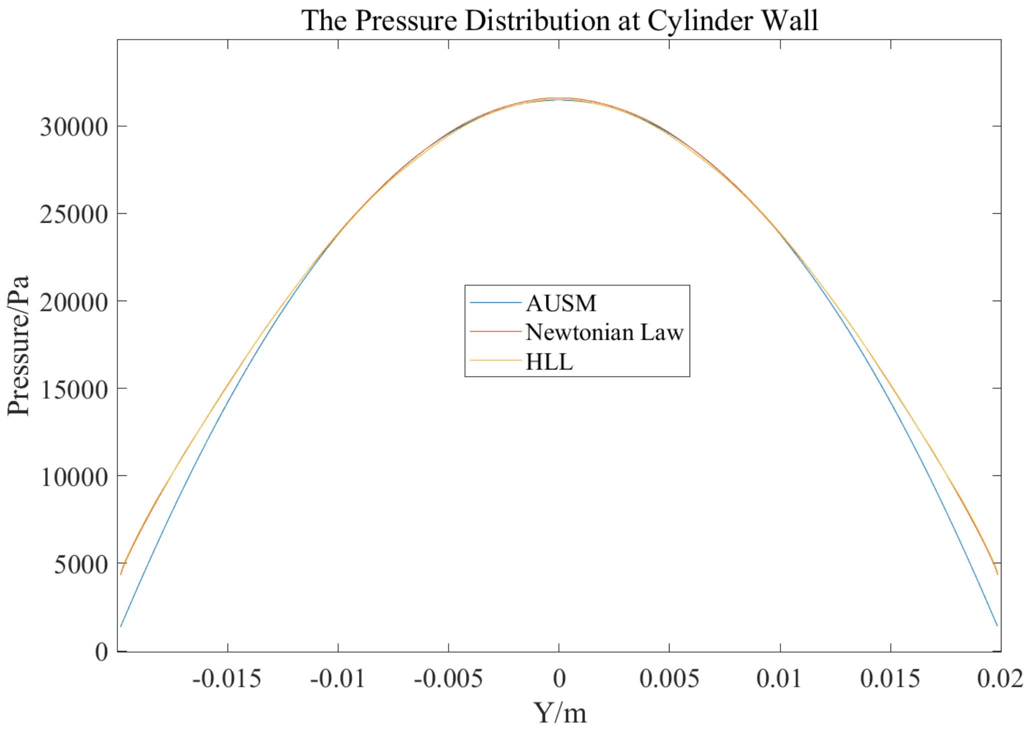


Figure 13 Comparison of Empirical and Numerical Pressure Distribution

(4) Summary:

Whether using the Riemann solver in HLL or AUSM scheme, the comparison of Figure9 - Fiugre13 shows that the numerical results of the FVM for the supersonic flow problem with a 2-D blunt cylinder body present very good results. The pressure distribution has only a slight inaccuracy at the left and right outlet boundaries, while the empirical shock shape fits the numerical solution well. The difference between HLL and AUSM scheme can be summarized as follows.

(a) Complexity and Cost: HLL is simpler and generally faster, making it suitable for problems where computational resources are limited. The AUSM family, being more complex, may require more computational effort but provides higher accuracy.

(b) Numerical Dissipation: HLL scheme may have more dissipation error than AUSM scheme when the grid becomes complex, and the slope of the wall varies greatly.

(c) Character Capture: HLL is used to 2 shock wave solution to approach Riemann problem, while AUSM uses a more detailed splitting of fluxes, which can capture more complex flow features, but AUSM family like AUSM+, AUSM-PW scheme needs more high order of time differential scheme such 4th order RK and MacCormack to fit the 2nd order space differential scheme.

Appendix

The source codes as well as the original Figure for the images can be found in the zip package.