A Discontinuous Galerkin program for the compressible Euler equations

Karthik Reddy Lyathakula

North Carolina State University, Raleigh, United States

Abstract

The objective of this project is to develop a program to numerically solve the 2D compressible Euler equations using the discontinuous Galerkin (DG) finite element method on unstructured grids. The code has been written using the C++ compiler in the Linux environment. The Euler equations are solved using three methods: DGP0(finite volume), DGP0 reconstructed using least squares, and P1 methods. The order of accuracy of the methods is determined by simulating flow across a bump with four meshes with increasing mesh elements. Finally, the program is tested on transonic flow over a NACA-0012 airfoil and supersonic flow inside a channel.

1. Introduction

Finite volume methods are extensively used to solve fluid dynamics problems. In the finite volume method, the flow variables are constant for each cell and this limits the accuracy. To increase the order of accuracy of the numerical simulation, Discontinues Galerkin Methods(DG) can be used or finite volume methods can be augmented by reconstruction to active higher order accuracy. In this project, Euler equations are solved for 2D flow problems to test the capability of reconstruction and DG methods.

2. Governing equaitons

In this project Euler equations are solved for 2d problems. The Euler equations in indicial notation is given by:

$$\frac{\partial U_i}{\partial t} + \frac{\partial F_i}{\partial x_j} = 0 \tag{1}$$

where,

$$\mathbf{U_i} = \begin{bmatrix} \rho \\ \rho u_i \\ \rho E \end{bmatrix} \quad \mathbf{F}_i = \begin{bmatrix} \rho u_j \\ \rho u_i u_j \\ u_i(E+p) \end{bmatrix}$$

U is the vector of the unknowns (i.e. density, velocities, and total energy), and **F** are the flux vectors. The pressure is given by

$$p = (\gamma - 1)\rho(E - \frac{1}{2}v_j v_j) \tag{2}$$

for two dimensional problem, the Euler equations will be

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}_x}{\partial x} + \frac{\partial \mathbf{F}_y}{\partial y} = \mathbf{0} \tag{3}$$

where,

$$\mathbf{U} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho E \end{bmatrix} \quad \mathbf{F}_x = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ u(E+p) \end{bmatrix} \quad \mathbf{F}_y = \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ v(E+p) \end{bmatrix}$$

3. Discontinous Galerkin method

Discontinuous Galerkin methods are a subset of Galerkin methods (Ref to project 1), where the basis function is same as the weigh function, but the basis function is continuous only inside each finite dimensional subspace(each mesh cell). The weak formulation for eq 1 is given by

$$\begin{gathered} \int \frac{\partial U_h}{\partial t} B d\Omega + \int \frac{\partial F_i(U_h)}{\partial x_i} B d\Omega = 0 \\ \int \frac{\partial U_h}{\partial t} B d\Omega + \int \Big(\frac{\partial (B F_i(U_h))}{\partial x_i} - F_i(U_h) \frac{\partial B}{\partial x_i} \Big) d\Omega = 0 \end{gathered} \tag{4}$$

applying gauss divergence theorem gives

$$\int_{\Omega_e} \frac{\partial U_h}{\partial t} B d\Omega + \int_{\Gamma_e} B F_i(U_h) n_i d\Gamma - \int_{\Omega_e} F_i(U_h) \frac{\partial B}{\partial x_i} d\Omega = 0 \tag{5}$$

here B is the weight function. The unknown U_h for each subspace can be approximated by different types of basis functions like Lagrarian, Taylor basis functions or legandary. In this project, Taylor basis are used which is given by

$$\begin{gathered} U_h = U_c(t) + \frac{\partial U(t)}{\partial x}\Big|_c (x - x_c) + \frac{\partial U(t)}{\partial y}\Big|_c (y - y_c) + \frac{\partial^2 U(t)}{\partial x^2}\Big|_c \frac{(x-x_c)^2}{2} + \frac{\partial^2 U(t)}{\partial y^2}\Big|_c \frac{(y-y_c)^2}{2} \\ + \frac{\partial^2 U(t)}{\partial x \partial y}\Big|_c \frac{(x-x_c)(y-y_c)}{2} \end{gathered} \tag{6}$$

the unknowns in the Taylor basis are the cell averaged variables and their derivatives at the center of the cells regardless of the element shapes. Normalized Taylor basis function is given by

$$U_h = \overline{U} B_1 + U_x B_2 + U_y B_3 + U_{xx} B_4 + U_{yy} B_5 + U_{xy} B_6 \tag{7}$$

where Basis functions (B_i) are given by

$$\begin{gathered} B_1 = 1 \qquad B_2 = \frac{x - x_c}{\Delta x} \qquad B_3 = \frac{y - y_c}{\Delta y} \qquad B_4 = \frac{(x-x_c)^2}{2\Delta x^2} - \frac{1}{\Omega_i} \int_{\Omega_i} \frac{(x-x_c)^2}{2\Delta x^2} d\Omega \\ B_5 = \frac{(y-y_c)^2}{2\Delta y^2} - \frac{1}{\Omega_i} \int_{\Omega_i} \frac{(y-y_c)^2}{2\Delta y^2} d\Omega \qquad B_6 = \frac{(x-x_c)(y-y_c)}{\Delta x \Delta y} - \frac{1}{\Omega_i} \int_{\Omega_i} \frac{(x-x_c)(y-y_c)}{\Delta x \Delta y} d\Omega \end{gathered} \tag{8}$$

here

$$\Delta x = \frac{x_{max} - x_{min}}{2} \qquad \Delta y = \frac{y_{max} - y_{min}}{2} \tag{9}$$

x_{max}, x_{min}, y_{max}, y_{min} are the maximum and minimum coordinates in the cell Ω_i in x and y directions respectively.

4. Fluxes

In the discretized Euler equation(5), the domain and boundary integration on each cell is evaluated by Gaussian quadrature numerical integration given by:

$$\mathbf{F}_{\Gamma_e} = \int_{\Gamma_e} \mathbf{F}_k \cdot \mathbf{n}_k B_i \, d\Gamma = \frac{\Gamma_e}{2} \sum_{i=1}^{ngauss} w_i f_i B_i \tag{10}$$

$$\mathbf{F}_{\Omega_e} = \int_{\Omega_e} \mathbf{F}_k \frac{\partial B_i}{\partial x_k} \, d\Omega = \Omega_e \sum_{i=1}^{ngauss} w_i f_i \frac{\partial B_i}{\partial x_k} \tag{11}$$

. The number of gauss points used to evaluate the fluxes depends on the order of the fluxes.

The flux at the boundaries of a cell is not continuous and there are different flux spitting schemes like Roe's method, Van Leer and Stegar warming method. In this project Vanleer method has been used to calculate the flux at boundaries (Eq. 10). For the sake of brevity, Van Leer's formulation has not been included in this report.

5. DGP0 Approximation

In DGP0 method, the conserved variable is approximated by one basis function and the approximation will become first order cell-centered finite volume scheme. In DGP0 the Eq. 5 will become

$$\frac{d}{dt} \int_{\Omega_e} U_h d\Omega + \int_{\Gamma_e} F_i(U_h) n_i d\Gamma = 0 \tag{12}$$

the fluxes on the boundaries for P0 is evaluated using one gauss point.

5.1. Reconstruction for $DG(P_0)$ (FVM)

The accuracy of the finite-volume solver can be increased by using reconstructed values at the cell-faces to calculate the fluxes, rather than the cell-centered average values. The values at the cell-center are extrapolated to the faces using the *least-squares* reconstruction. The least square reconstruction is given by

$$\begin{bmatrix} \sum_j^n w_j^2 \Delta x_j^2 & \sum_j^n w_j^2 \Delta x_j \Delta x_i \\ \\ \sum_j^n w_j^2 \Delta x_j \Delta x_i & \sum_j^n w_j^2 \Delta y_j^2 \end{bmatrix} \begin{bmatrix} \frac{\partial u}{\partial x}|_i \\ \\ \frac{\partial u}{\partial y}|_i \end{bmatrix} = \begin{bmatrix} \sum_j^n w_j^2 \Delta x_j \Delta u_j \\ \\ \sum_j^n w_j^2 \Delta x_j \Delta u_j \end{bmatrix}$$

where

$$\begin{aligned}\Delta x_j &= x_j - x_i \\ \Delta y_j &= y_j - y_i \\ \Delta u_j &= u_j - u_i\end{aligned} \tag{13}$$

here n are the number of cells surrounding the current cell(i). The basic idea of the reconstruction is to evaluate the first derivative of the unknown in the cell, so that a linear-polynomial data can be retrieved from the original piecewise-constant data inside each element. The weighting factor in these equations is,

$$w_j = \|\overrightarrow{x_j} - \overrightarrow{x_i}\|^{-P} \tag{14}$$

where $\overrightarrow{x_j}$ is the centroid of the neighboring cell and $\overrightarrow{x_i}$ is the centroid of the cell over which the derivative is required. In this project P is taken as -1 which makes the weight 1. Once the derivative is calculated by solving the simultaneous equations, it is used to reconstruct the value of the unknown quantity at the cell-face using a second-order accurate Taylor-series expansion given by:

$$u_i(x,y) = \overline{u_i} + \frac{\partial u}{\partial x}\Big|_i (x - x_i) + \frac{\partial u}{\partial y}\Big|_i (y - y_i) \tag{15}$$

This reconstructed value is used to calculate the fluxes from 10.

5.2. Slope limiters

Reconstruction in DG methods leads to oscillations in discontinuous flow fields. This causes the solution to diverge. To solve this problem, special functions are needed to limit either the flux or the slopes ($\frac{\partial \mathbf{U}}{\partial x_i}$) at the interfaces. Van Albada's slope limiter has been used here. It is easy to incorporate into the $DG(P_0)$ solver, and can be switched-off when not needed (for example, for subsonic flows). The Van Albada limiter function is,

$$\phi_i = max\left\{0, \frac{2(\Delta_i^-)(u_j - u_i) + \epsilon}{(\Delta_i^-)^2 + (u_j - u_i)^2 + \epsilon}\right\} \tag{16}$$

where,

$$\Delta_i^- = 2(\vec{\nabla} u_i) \cdot (\overrightarrow{x_j} - \overrightarrow{x_i}) - (u_j - u_i) \tag{17}$$

and similar relations for ϕ_j and Δ_j^+. The slopes are limited as,

$$u_i^+ = u_i + \frac{\phi_i}{4}\left[(1 - k\phi_i)\Delta_i^- + (1 + k\phi_i)(u_j - u_i)\right] \tag{18}$$

These values are then used to calculate the fluxes. The limiters have been applied on the primitive variables i.e. ρ, u, v and p since the performance of limiters is known to be better when applied to the primitive variables rather than the conserved variables.

6. DGP1

In DGP1, the conserved variable is approximated by linear function

$$U_h = \overline{U}B_1 + U_x B_2 + U_y B_3 \tag{19}$$

here $\overline{U}$, U_x, U_y will be the unknowns. The equations for the DGP1 will be

$$\frac{d}{dt}\int_{\Omega_e} U_h d\Omega + \int_{\Gamma_e} F_i(U_h) n_i d\Gamma = 0 \tag{20}$$

$$\begin{bmatrix} \int B_2 B_2 d\Omega & \int B_2 B_3 \\ \int B_2 B_3 d\Omega & \int B_3 B_3 \end{bmatrix} \begin{bmatrix} \frac{dU_x}{dt}|_i \\ \\ \frac{dU_y}{dt}|_i \end{bmatrix} + \begin{bmatrix} \int_\Gamma F_k n_k B_2 d\Gamma - \int_\Omega F_k \frac{\partial B_2}{\partial x_k} d\Omega \\ \\ \int_\Gamma F_k n_k B_3 d\Gamma - \int_\Omega F_k \frac{\partial B_3}{\partial x_k} d\Omega \end{bmatrix} = 0$$

For each conservative variable, three equations are solved to get three unknowns. To evaluate fluxes on the boundaries, two gauss points has been used and for the domain integral three gauss points has been used.

7. Boundary conditions

7.1. Slip wall

Slip wall boundaries condition has been used for all the walls. The equations for the ghost cells and ghost values are given by

$$\begin{aligned} \rho_g &= \rho_{ie} \\ (\rho E)_g &= (\rho E)_{ie} \\ (v_t)_g &= (v_t)_{ie}(v_n)_g = -(v_n)_{ie} \end{aligned} \tag{21}$$

here sibscript 'ie' indicates the value in the internal cell.

7.2. Far Field Boundary condition

For far field boundaries, characteristic boundary conditions based on Mach number(M_n) has been used. The Mach number is given by

$$M_n = \frac{v_{ie}}{c} \tag{22}$$

here 'c' is the speed of sound.

7.2.1. Supersonic Inflow($M_n < -1$)

All quantites at the ghost state and cells is assigned to free stream condition

$$\overrightarrow{U}_g = \overrightarrow{U}_\infty \tag{23}$$

7.2.2. Subsonic Inflow($-1 < M_n < 0$)

The values of conserved variables at ghost cells, ghost states is calculated by solving following equations

$$\begin{aligned} v_{ng} - \frac{2.0c_g}{\gamma - 1} &= v_{n\infty} - \frac{2.0c_\infty}{\gamma - 1} \\ v_{tg} &= v_{t,\infty} \\ \frac{p}{\rho^\gamma} &= \frac{p_\infty}{\rho_\infty^\gamma} \\ v_{ng} + \frac{2.0c_g}{\gamma - 1} &= v_{n,ie} + \frac{2.0c_{ie}}{\gamma - 1} \end{aligned} \tag{24}$$

7.2.3. Subsonic Outflow($0 < M_n < 1$)

The values of conserved variables at ghost cells, ghost states is calculated by solving following equations

$$\begin{aligned} v_{ng} - \frac{2.0c_g}{\gamma - 1} &= v_{n\infty} - \frac{2.0c_\infty}{\gamma - 1} \\ v_{tg} &= v_{t,ie} \\ \frac{p}{\rho^\gamma} &= \frac{p_{ie}}{\rho_{ie}^\gamma} \\ v_{ng} + \frac{2.0c_g}{\gamma - 1} &= v_{n,ie} + \frac{2.0c_{ie}}{\gamma - 1} \end{aligned} \tag{25}$$

7.3. Supersonic Outflow($M_n > 1$)

All quantities at the ghost state and cells is assigned to values calculated from internal cells.

$$\overrightarrow{U}_g = \overrightarrow{U}_{ie} \tag{26}$$

8. Time integration

The time-derivative in 5 has been integrated by a 3-stage Runge-Kutta scheme, with an option of using a 1-stage Runge-Kutta scheme. The 3-stage Runge-Kutta scheme is given by,

$$\begin{array}{lcl} U^{(1)} & = & U^n + \Delta t M^{-1} R(U^n) \\ U^{(2)} & = & \frac{3}{4}U^n + \frac{1}{4}(U^{(1)} + \Delta t M^{-1} R(U^{(1)})) \\ U^{(3)} & = & \frac{1}{3}U^n + \frac{2}{3}(U^{(2)} + \Delta t M^{-1} R(U^{(2)})) \\ U^{n+1} & = & u^{(3)} \end{array} \tag{27}$$

Note that the RHS flux matrix is calculated on the basis of the solution at the previous RK stage and the boundary conditions are applied after each RK stage.

9. Convergence criterion and error measure

The convergence is decided on the basis of the residuals at each iteration as:

$$\parallel u^{n+1} - u^n \parallel_{L2} = \sqrt{\sum_i \Omega_i (u_i^{n+1} - u_i^n)^2} \tag{28}$$

When the residuals reach 10^{-6}, convergence is assumed to have been achieved. The error, as we do not have a analytical solution to the Euler equations is expressed in terms of entropy production as:

$$\epsilon_i = \frac{s_i - s_\infty}{s_\infty} \tag{29}$$

where s is the entropy, given as:

$$s_i = \frac{p}{\rho^\gamma} \tag{30}$$

The L2-norm of the error has been calculated as:

$$\parallel \epsilon \parallel_{L2} = \sqrt{\sum_i \Omega_i \epsilon_i^2} \tag{31}$$

This error is used to determine the order of convergence of the methods used here.

10. Results and Discussion

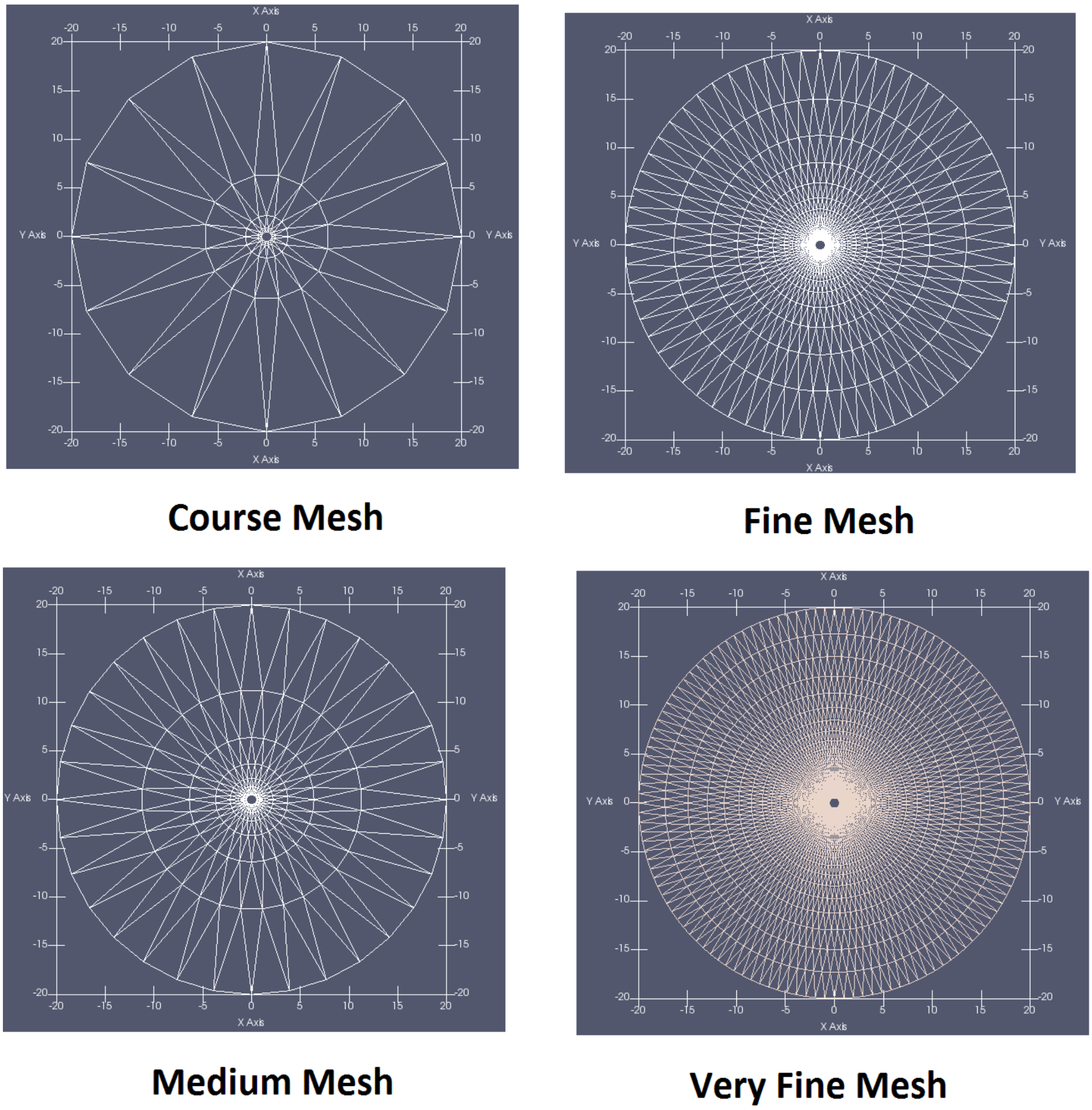
A generalized C++ program is developed to solve the Euler equations using P0, reconstructed P0P1, and reconstructed P0P1 with a limiter and P1 method. The choice of method to solve the Euler equations can be controlled by an input file. The developed program is tested on three test cases using all the methods. First test case is subsonic flow past a cylinder, second case is internal flow through bump and third case is flow past an airfoil.

10.1. Subsonic flow past a circular cylinder (M=0.38)

Initial simulation are conducted on flow past a cylinder, to determine the order of accuracy for four methods, by solving the Euler equations on different grids with decreasing characteristic mesh sizes(Fig. 1). The Mach number used for this case is subsonic(Mach=0.38) and CFl of 0.2 is used for all runs. The order of accuracy is determined by the relation between total entropy production in the domain (Eq. 31), and characteristic grid size given by:

$$\begin{gathered} ||E||_{L^2} = C * h^{\alpha} \\ log10(||E||_{L^2}) = \alpha * log10(h) + log10(C) \end{gathered} \tag{32}$$

the coefficient α determines the order of accuracy of the discretization method. Ideally, the entropy production in each cell must be zero because there are no losses in the system. However, as numerical methods are based on approximations which will induce some numerical dissipation in the solution and hence the total entropy production in the system is finite.



Course Mesh

Fine Mesh

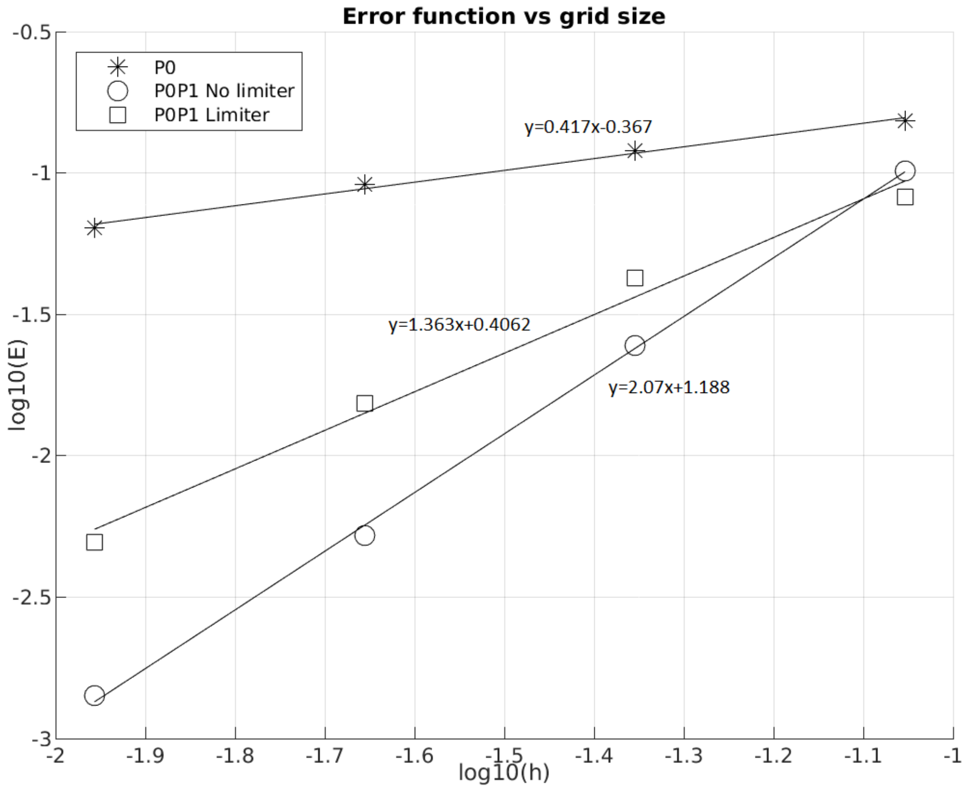
Medium Mesh

Very Fine Mesh

(a) Course Mesh

Figure 1: Meshes for determining the order of accuracy

Fig. 2 shows the comparison of error function vs the characteristic grid sizes for different methods. It shows that the entropy production is very high using P0(finite volume) method as there is lot of numerical dissipation and hence the order of accuracy is less than one. Using reconstructed P0P1 method without the limiter, the order of accuracy increases to second order method. It was observed from the simulations that there are lot of oscillations in the solution without the limiter and it induces convergence issues when solving for higher Mach numbers. To avoid it, Van Alberta limiter has been used and it was observed that the order of accuracy has been decreased because of limiting the reconstruction.

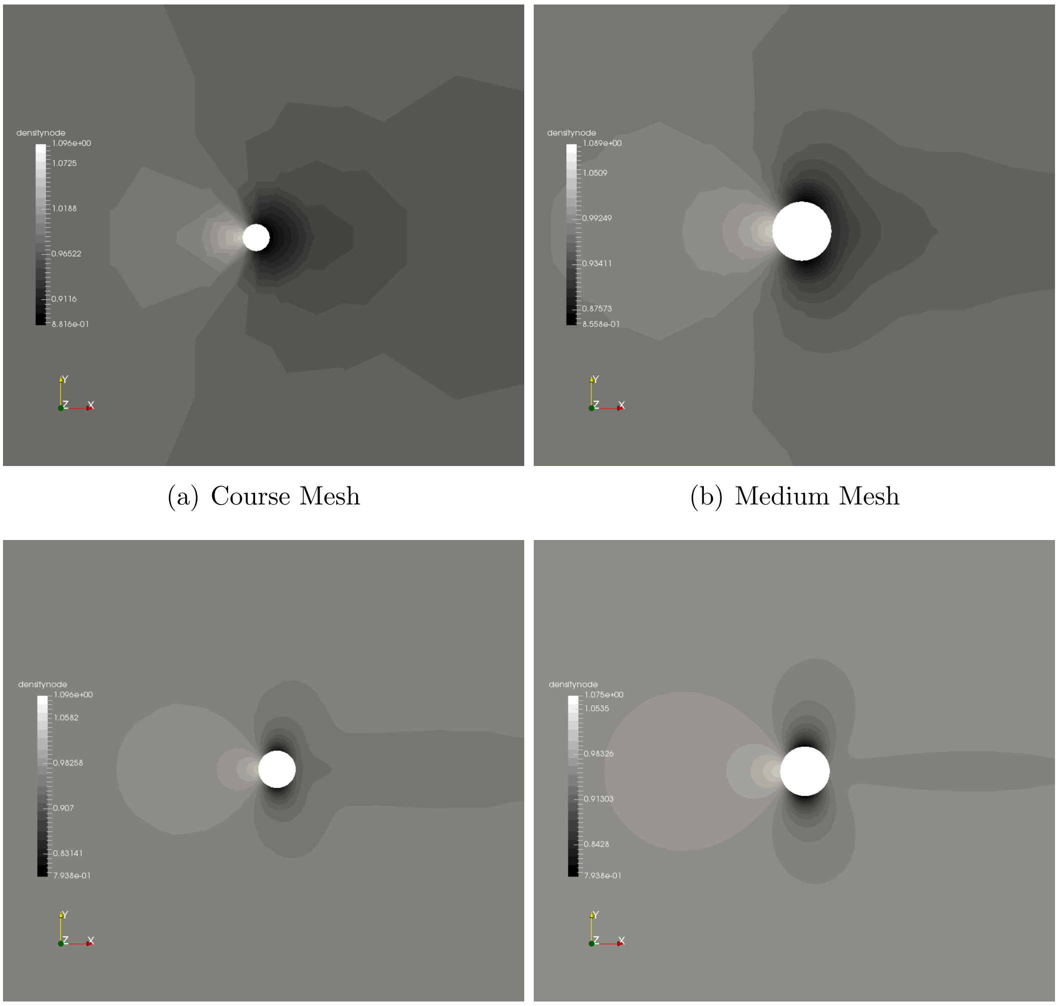


(a) Course Mesh

Figure 2: Meshes for determining the order of accuracy

To improve the order of accuracy without using limiter, P1 method has been implemented but second order accuracy is not achieved because of bug in the code. However, the required order of accuracy is achieved using reconstruction.

Fig 3, Fig. 4, Fig. 5 shows the comparison of density, pressure and Mach number for four different meshes using P0 method. It can be observed that with mesh refinement, the pressure, density and Mach Number becomes smooth and symmetric about y axis, which is expected for subsonic velocities. However, the solution is not perfectly symmetric even in the fine mesh because the P0 method is dissipative.



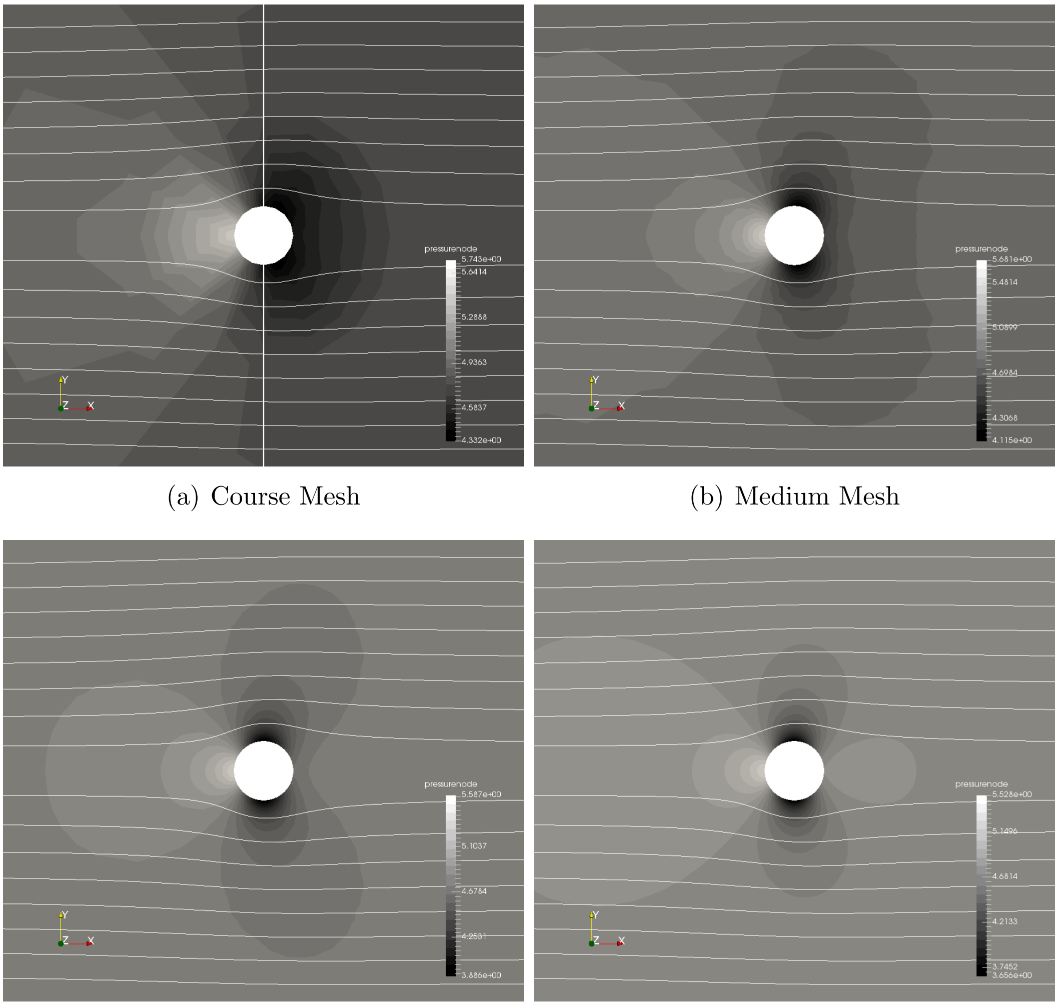
(a) Course Mesh

(b) Medium Mesh

(c) Fine Mesh

(d) Very fine Mesh

Figure 3: Comparision of density contours for different meshes



(a) Course Mesh

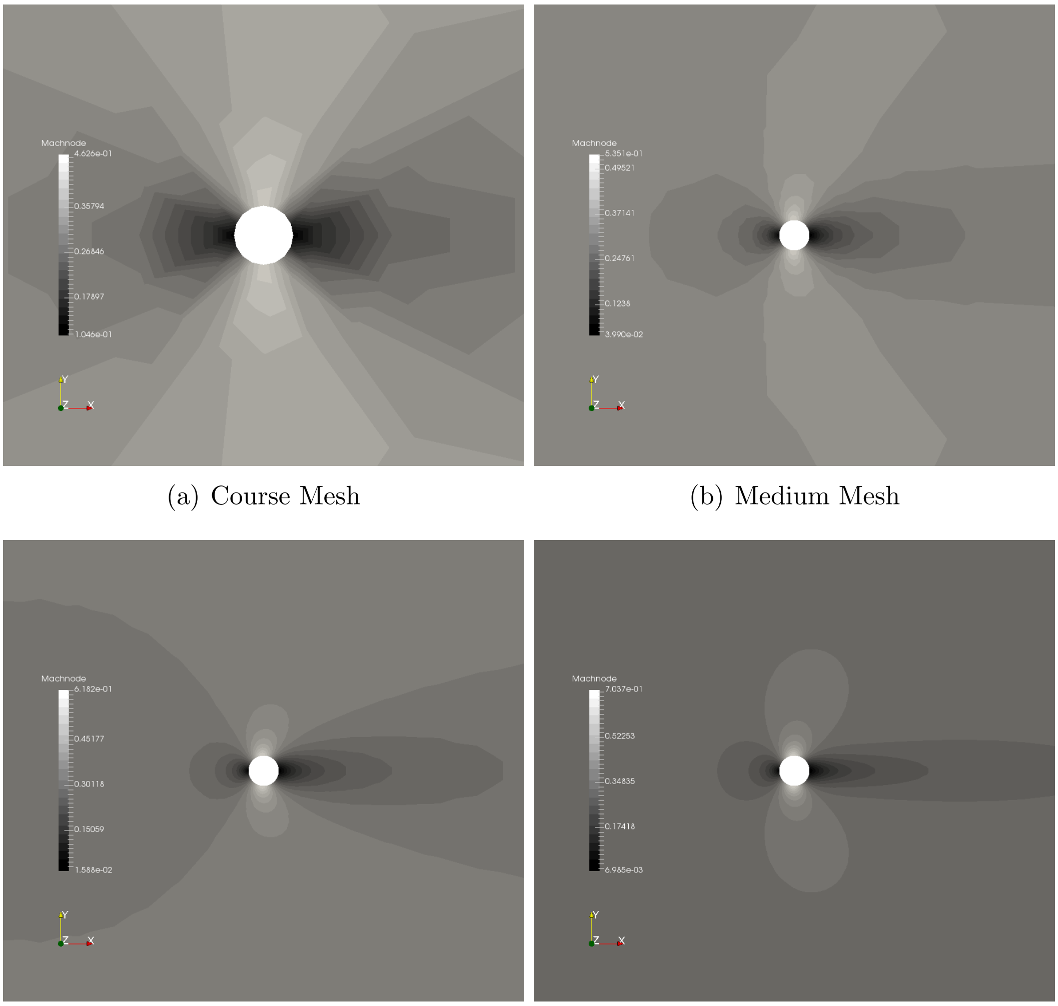
(b) Medium Mesh

(c) Fine Mesh

(d) Very fine Mesh

Figure 4: Comparision of pressure contours for different meshes

The maximum Mach number for flow past a cylinder is 0.7 for fine mesh and it occurs at the top of the cylinder as expected. It can be also observed that the Maximum Mach number increases with mesh refinement.



(a) Course Mesh

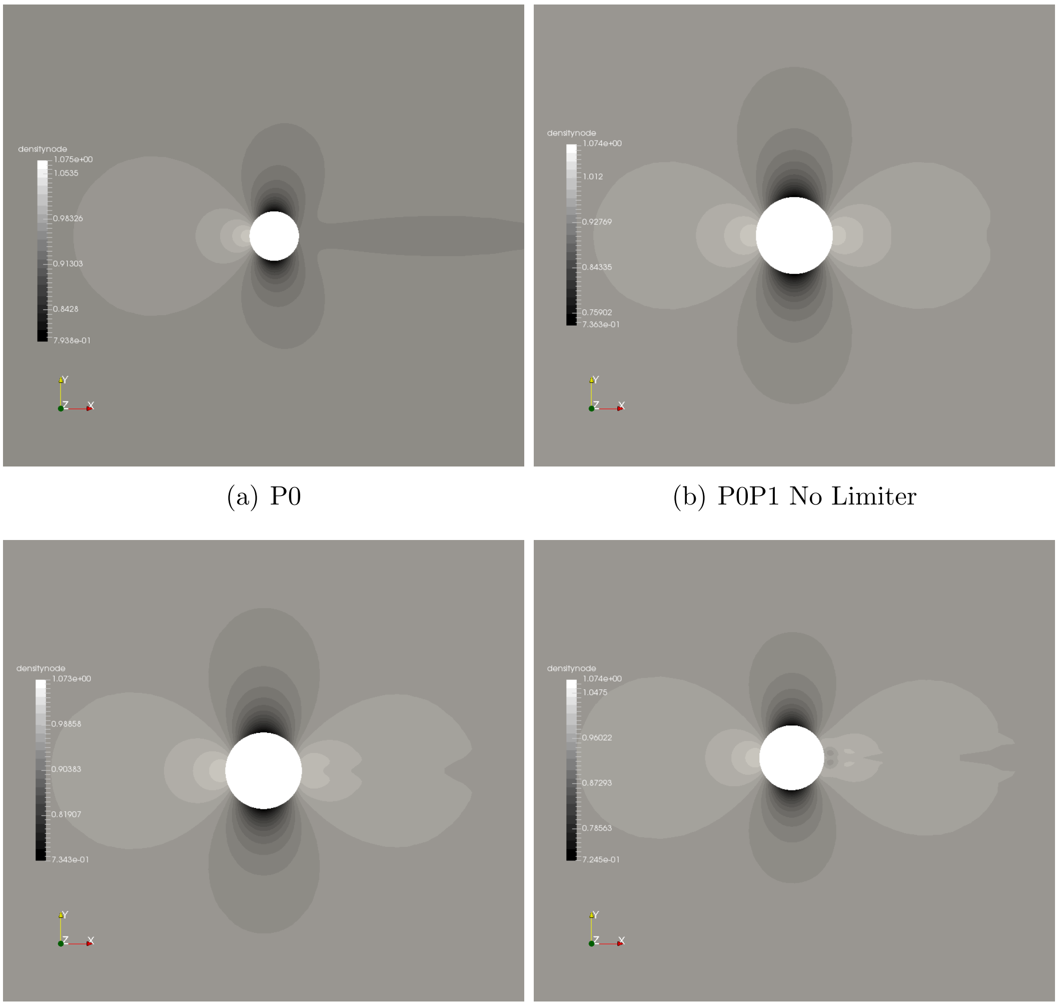
(b) Medium Mesh

(c) Fine Mesh

(d) Very fine Mesh

Figure 5: Comparision of Mach Number for differen meshes

Fig. 7, Fig. 8 and 6 shows the comparison of pressure, Mach number and density on very fine mesh for P0, reconstructed P0P1 without limiter, reconstructed P0P1 with limiter and P1 methods. It can be observed that the solution is almost symmetric about y axis with P0P1 reconstruction without a limiter as the order of accuracy is two and this is expected for a subsonic flow. The P0P1 reconstruction with limiter decreases the accuracy and it is evident from unsymmetrical pressure contours about y axis.



(a) P0

(b) P0P1 No Limiter

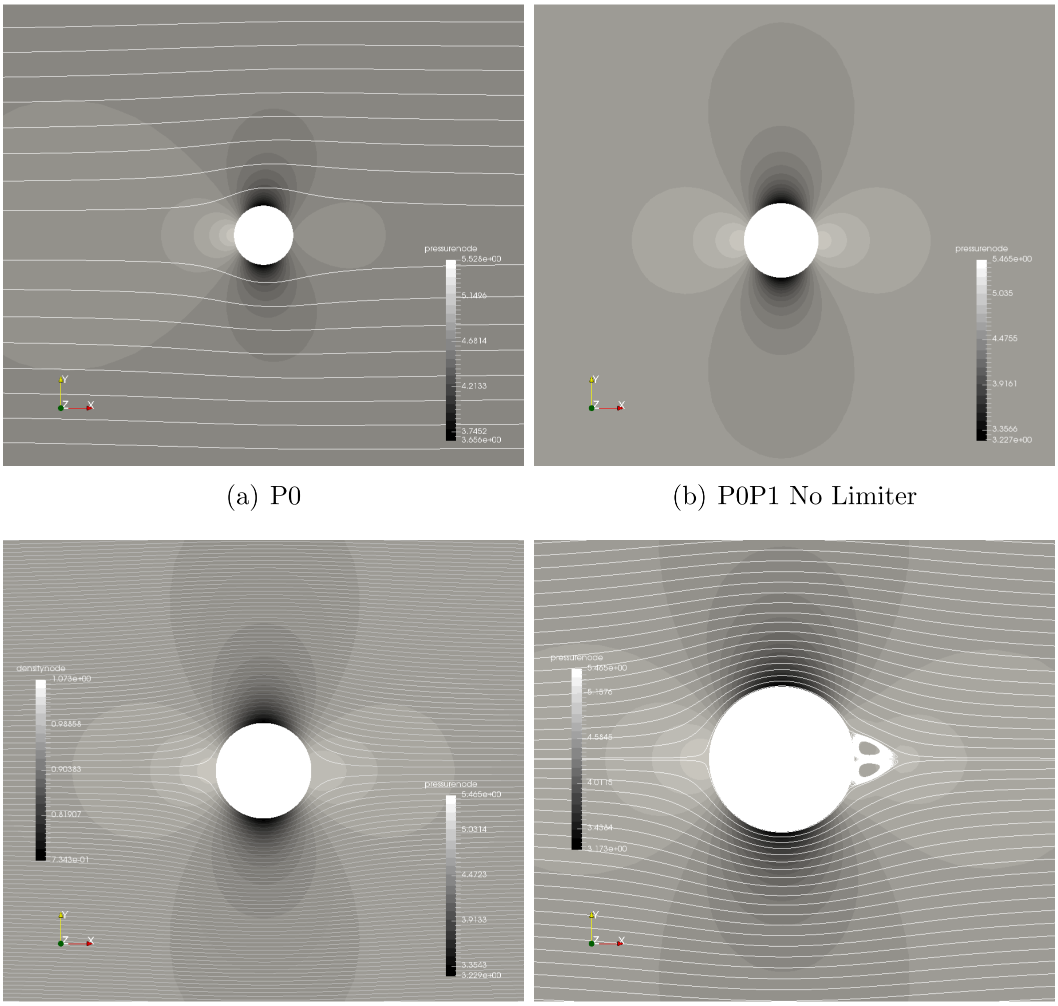
(c) P0P1 Limiter

(d) P1

Figure 6: Comparision of density contours for different methods

10.1.1. Issues with P1 method

The order of accuracy for the P1 method is determined to be 1.3 and the reason for this is because of numerical dissipation in the solution caused by bug in the program. This is evident from the circulation behind the cylinder in the streamline plot of P1 solution which generally occurs in case of viscous flows. It was also observed that there is large entropy production near the circulation which is caused by numerical dissipation. This concludes that the P1 method required to be reviewed to ensure correct results which will improve the order of accuracy.



(a) P0

(b) P0P1 No Limiter

(c) P0P1 Limiter

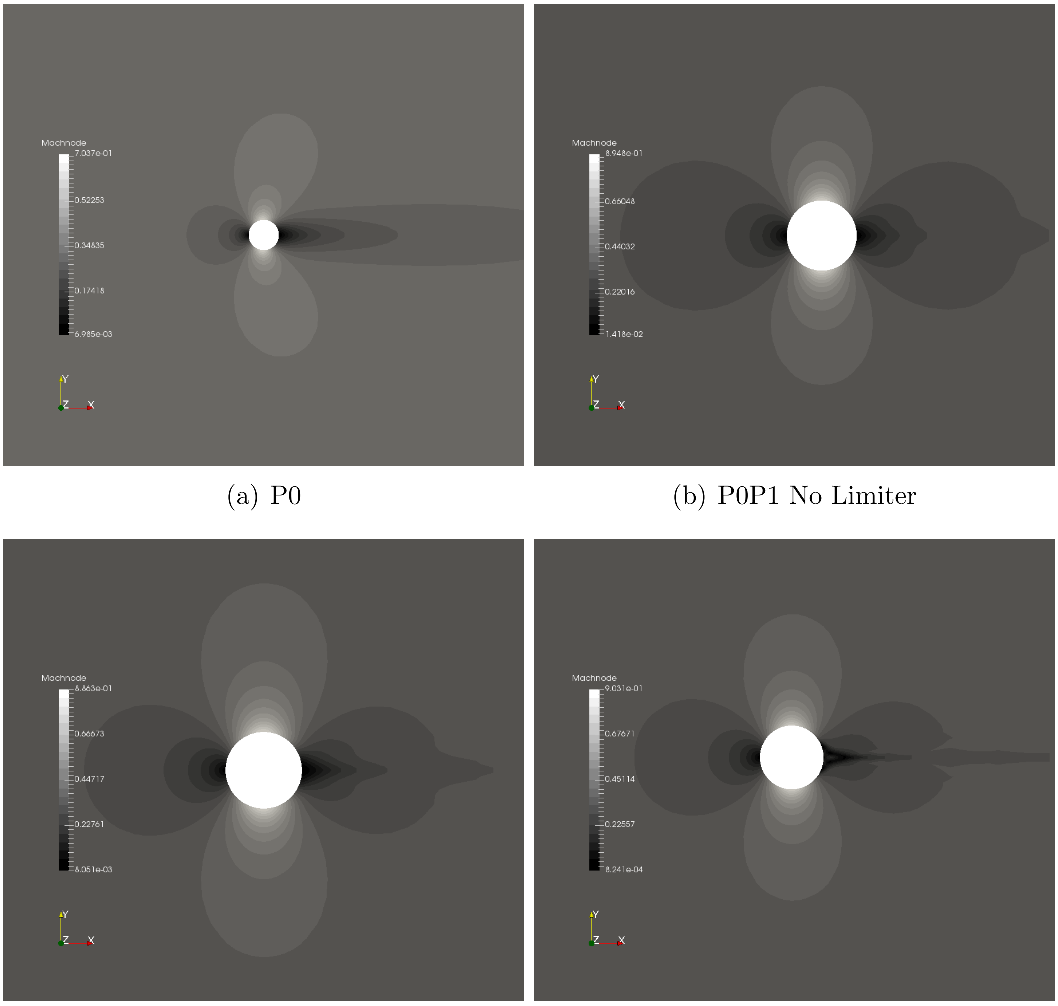
(d) P1

Figure 7: Comparision of pressure contours for different methods

The Mach Number plot shows that the contours are symmetric about y axis without limiter as it is second order accurate and unsymmetrical in all other cases because of numerical dissipation caused by lower order of accuracy.

10.1.2. Convergence plot

Fig. 9 shows the plot of Residues vs the iteration number. The relative residue is the ratio of absolute residue at current iteration and absolute residue after first iteration. The solution is considered to be converged when the relative residue is 1e-6.



(a) P0

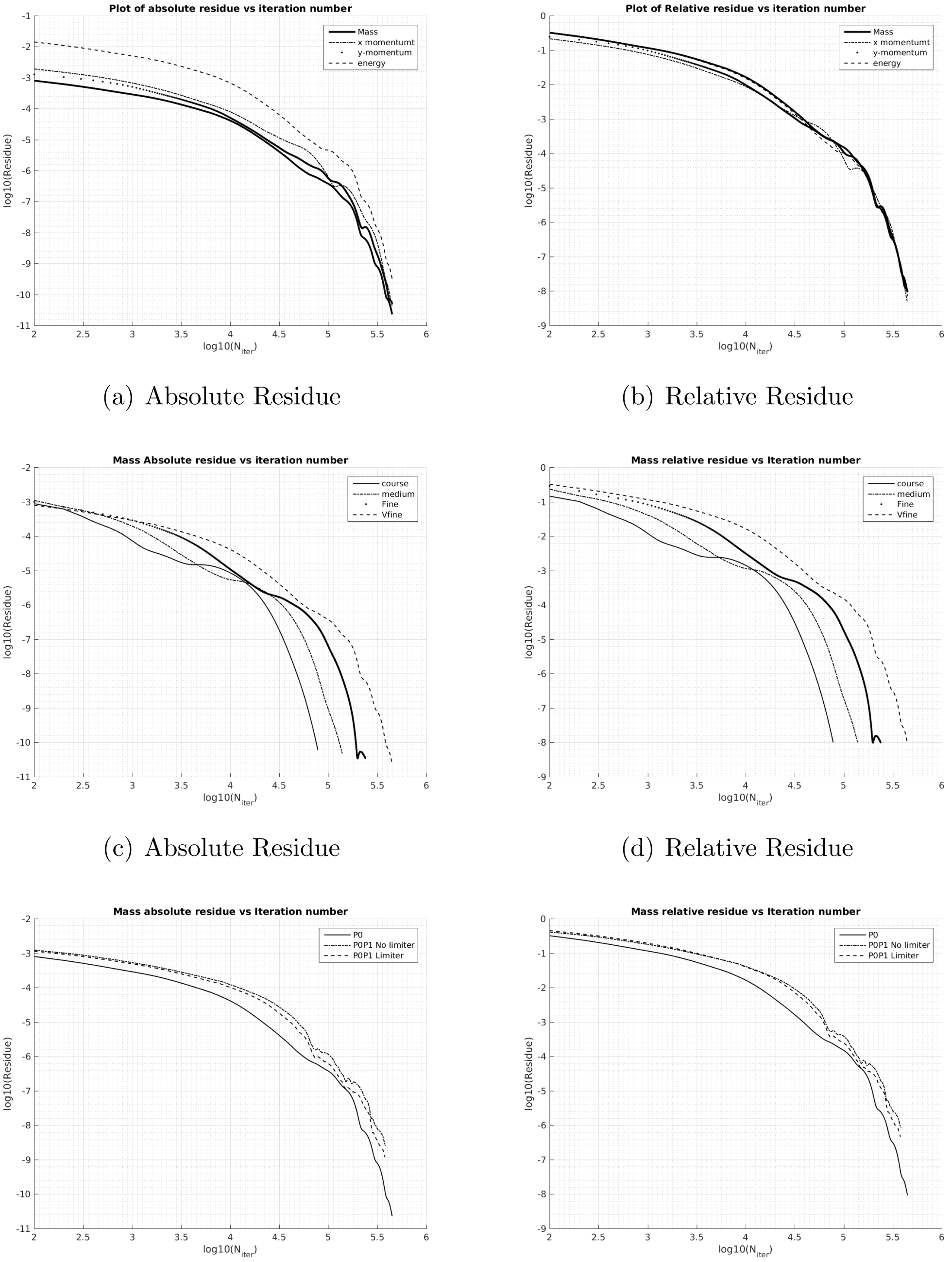
(b) P0P1 No Limiter

(c) P0P1 Limiter

(d) P1

Figure 8: Comparision of Mach number for different methods

It can be observed from the plot that relative residue of continuity, momentum, energy follows almost similar trend with slightly more number of iteration for relative residue of mass to reach the tolerance (1e-6) and hence mass residue plots are compared for subsequent cases. Fig. 9.c and d shows the comparison of convergence plots for different meshes. It can be observed from the plot that the very fine mesh take more number of iterations to converge and absolute residue for all the meshes is around 1e-10. Fig 9.e and 9, shows the comparison of convergence plots for different methods on very fine mesh. It is observed that all the cases take similar number of iterations to converge.



(a) Absolute Residue

(b) Relative Residue

(c) Absolute Residue

(d) Relative Residue

(e) Absolute Residue

(f) Relative Residue

Figure 9: Comparision of Residues for flow past a cylinder

10.2. Transonic flow past NACA airfoil (M=0.85)

The second case is flow past an airfoil with free stream Mach number of 0.85 at 1 degree angle of attack. Fig. 10 shows the grid near the airfoil.

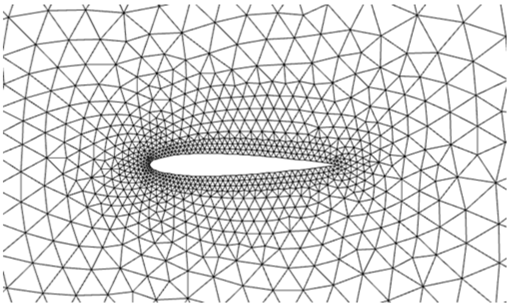
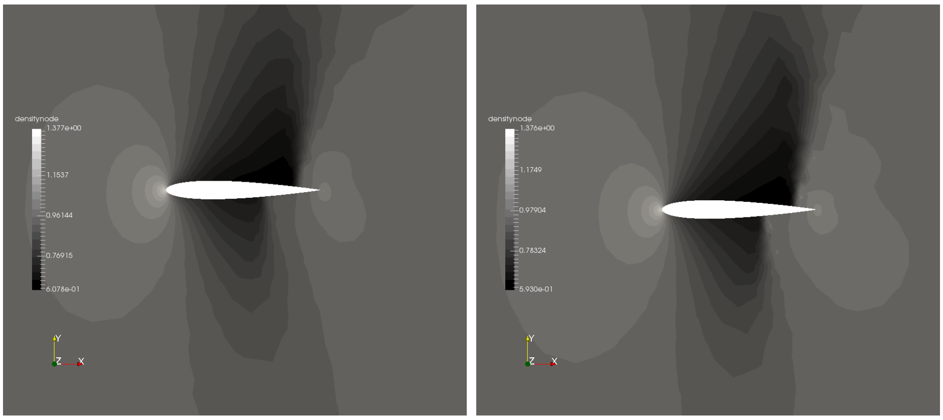


Figure 10: Comparision of flow inside internal flow

The simulation is conducted using P0, reconstructed P0P1 without limiter and with limiter. Fig. 11 and 12 shows the comparison of density, pressure and Mach number for all the three cases. It can be observed that the solution is not symmetric about the x axis because of finite angle of attack.

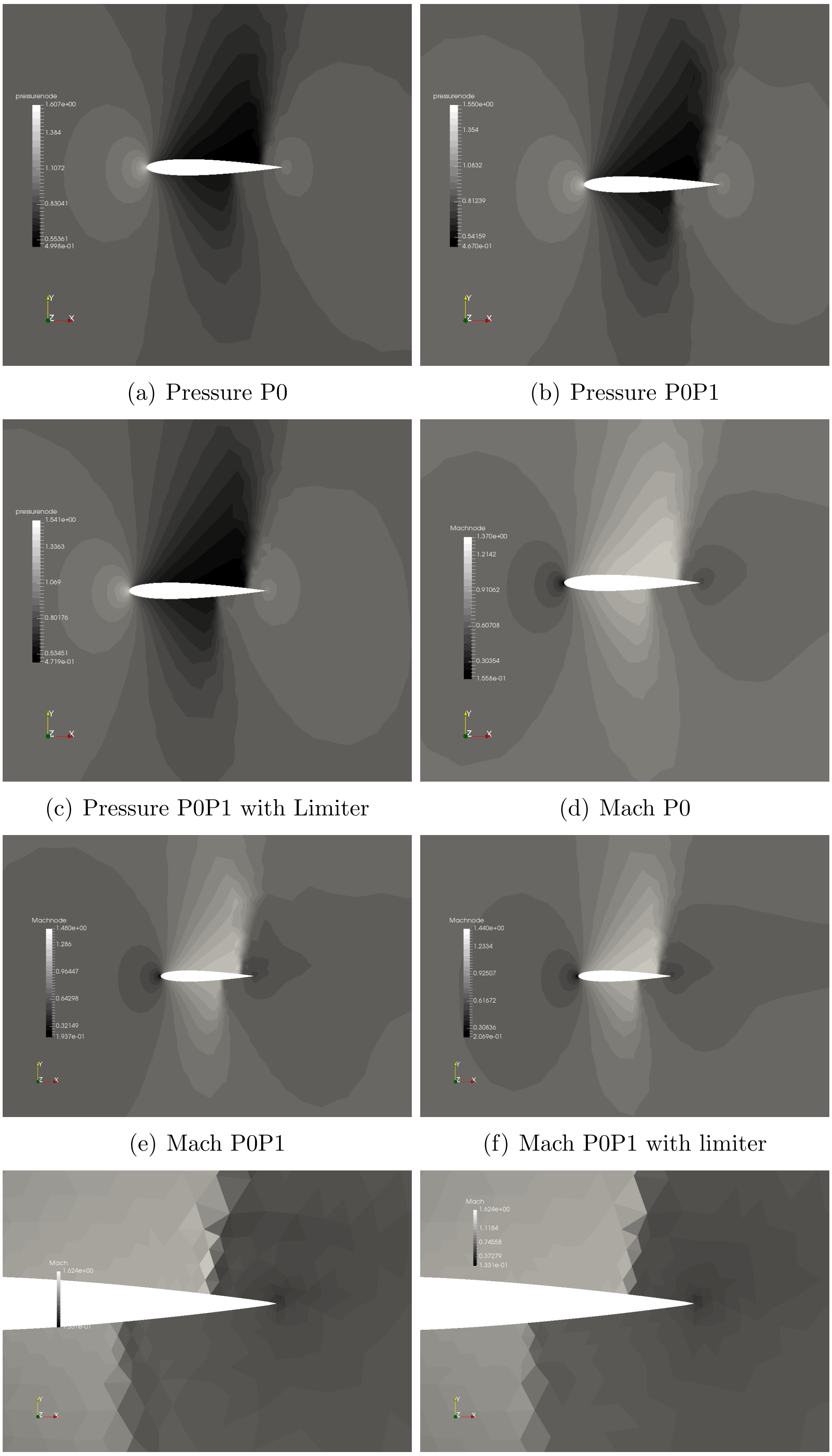
The contour plots shows that the density, pressure is maximum and Mach number is minimum at the nose of the airfoil. Pressure and density decreases over the airfoil whereas the Mach number increase due to decrease in cross-section. Finally there is a shock near the trailing edge of the airfoil where density, pressure increases and Mach number decreases. Comparing the solutions of all the methods it can be observed that shock is sharp in case of reconstruction compared to P0 Case. However, there are oscillation without the limiter and oscillations are reduced with the limiter. This is evident from the fig 12.g and 12.h.



(a) Density P0

(b) Density P0P1 with Limiter

Figure 11: Solution of NACA airfoil



(a) Pressure P0

(b) Pressure P0P1

(c) Pressure P0P1 with Limiter

(d) Mach P0

(e) Mach P0P1

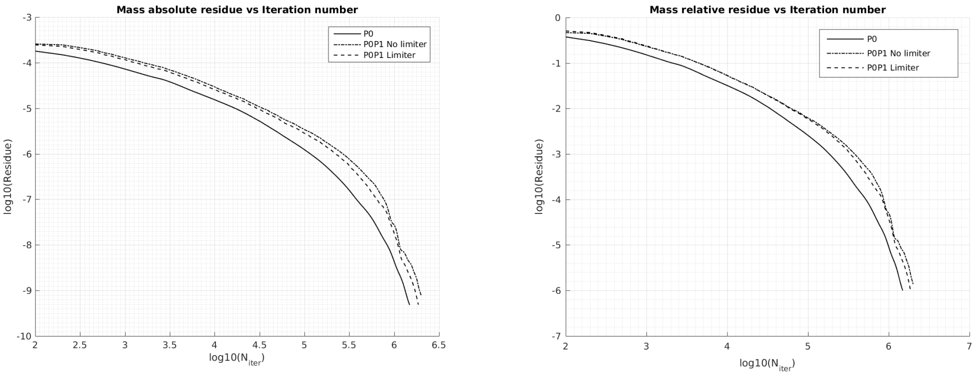
(f) Mach P0P1 with limiter

(g) Mach P0P1

(h) Density P0P1 with Limiter

Figure 12: Solution of NACA airfoil

Fig. 13 shows the convergence plot for all the three methods for airfoil. It is observed that the P0 takes least number of iterations and P1 with limiter takes most number of iterations.



(a) Absolute Residue

(b) Relative Residue

Figure 13: Comparision of Residues for Naca airfoil

10.3. Supersonic flow in a channel(M=1.1)

In the third case, internal flow over a circular bump with inlet Mach number of 1.1 is simulated with p0, reconstructed P0P1 without limiter and with limiter. Fig. 14 shows the mesh of the channel.

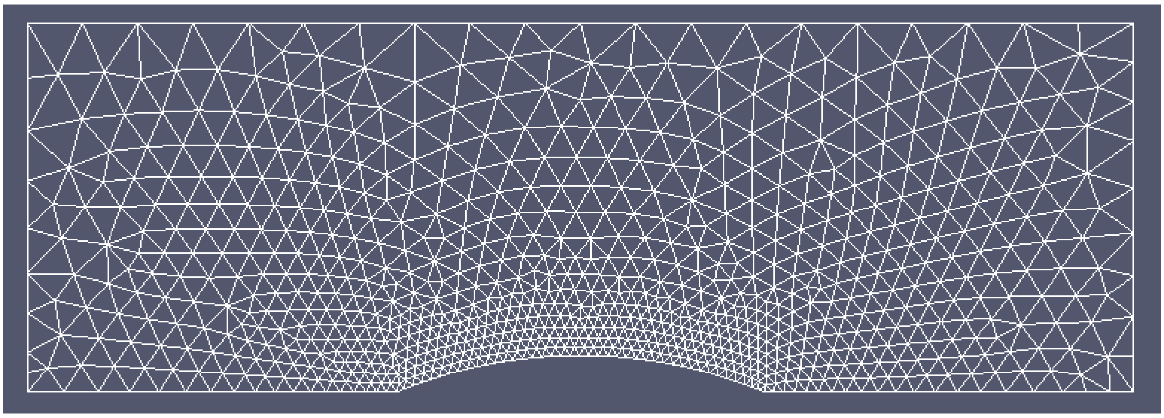
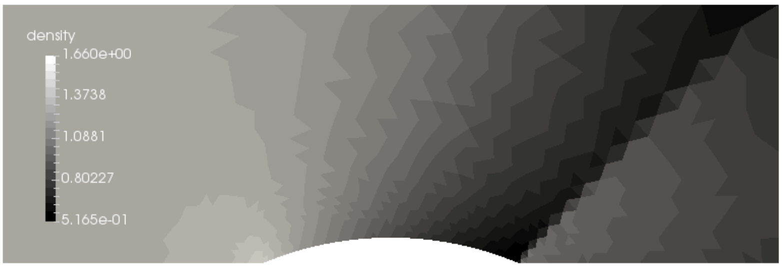


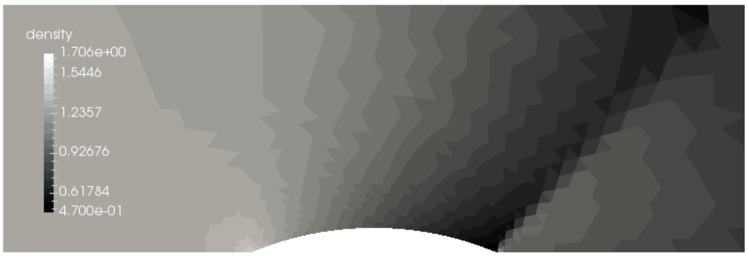
Figure 14: Comparision of flow inside internal flow

It is observed that the case without limiter does not converge because of the huge oscillation in solution for supersonic Mach number and hence the results without limiter is not reported. Fig. 15 shows the comparison of pressure, density and Mach number contours for internal flow with P0 and reconstructed P0P1 method with limiter. It can observed that the Mach number at the inlet is subsonic despite the supersonic inflow. This is because the shock wave occurs just before the entrance as there is decrease in cross-section at the bump.

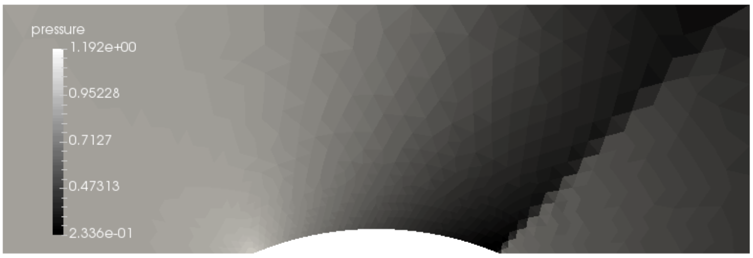
The Mach number continuously increase from the entrance to the bump, as the cross section is decreasing and the Mach number decreases downstream the bump followed by a shock at the corner because of the geometric singularity. Form the figure, it can be observed that the accuracy increases with reconstruction and sharp gradients can be seen across the shock. However, there are oscillation at the upper surface near the shock due to reconstruction despite using limiter.



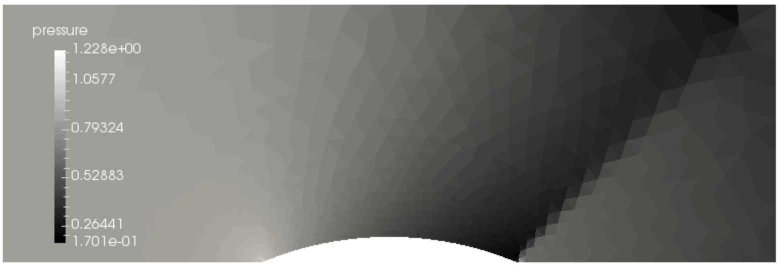
(a) P0



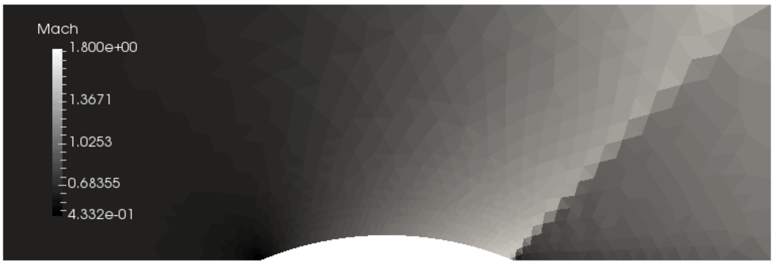
(b) P0P1 Limiter



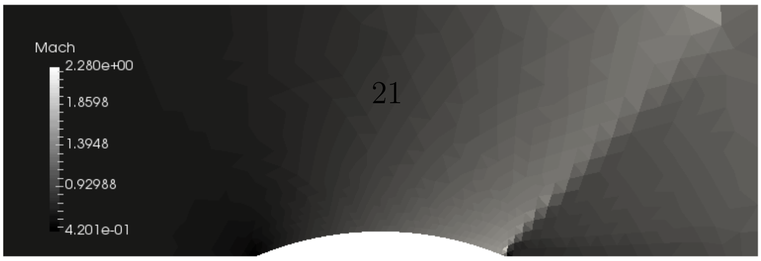
(c) P0



(d) P0P1 limiter



(e) P0



11. Conclusion

In conclusion, the C++ program to solve Euler equation is successfully tested on three different cases with four discretization methods. From the subsonic flow past a cylinder simulation, the order of accuracy for reconstructed P0(Finite Volume) method is determined to be first order, P0P1 method without the limiter is second order accurate. It was observed that the reconstruction induces oscillation for higher Mach numbers and to avoid it Van-Alberta limiter has been introduced. However the order of accuracy of reconstruction with limiter decreases. To improve the accuracy, P1 method has been implemented but the order of accuracy has not been achieved because of a bug in the program.In the second test case, program is successfully tested on transonic flow past an airfoil and results looks to be promising. Finally supersonic flow inside a channel has been modelled and the reconstructed case without limiter did not converge because of the oscillations in the solution. It is observed that the oscillations in the solutions can be reduced by using a limiter and it is found that the simulation converged successfully with limiter.

12. Future Work

In this project, P1 method has been implemented to test the capability of DG method but the accuracy found to be reduced due to a bug in the program. Implementing the DGP1 method would be promising and it would give the second order solution without the effect of numerical oscillations when using reconstruction for higher Mach number.