Development of General Compressible Flow solver

Ramkumar S.

The general Compressible flow solver is under development, as a sample case the following simulation(s) were done using the solver. The laminar viscous supersonic flow over flat plate at zero degree angle-of-attack is numerically computed using Finite difference method and full Navier-Stokes equations. Maccormack's predictor-corrector method is used for solution in this work, and can be extended as the code is made in modular way. Computation code is developed using Fortran programming language and the post-processing is done using ParaView. The book by John D. Anderson , "Computational Fluid Dynamics - basics with applications" is used as the main reference for the work.

I. Problem description

The following points describe the problem definition.

- Freestream Mach number is taken as 4.0 and the other parameters, density, pressure and temperature were taken to be the values at standard sea level.
- Plate length is taken to be 1e-5 m. and the domain height is taken to be five times the boundary layer height at the end of the plate, which comes to be around 7.6e-6 m. The grid size of 100X100 is taken for the computation.

A schematic image taken from the book is shown in the Figure 1.

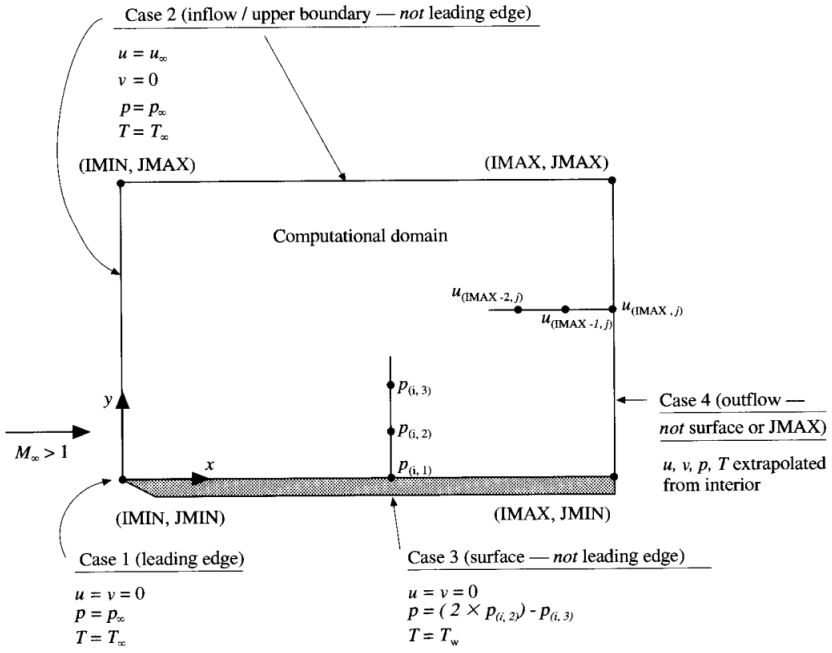


Fig. 1 Schematic diagram along with boundary conditions for the present problem, taken from the book

II. Computation procedure

The governing equations used in the work is given below.

$$\frac{\partial U}{\partial t} + \frac{\partial E}{\partial x} + \frac{\partial F}{\partial y} = 0$$

$$U = \begin{Bmatrix} \rho \\ \rho u \\ \rho v \\ E_t \end{Bmatrix}$$

$$E = \begin{Bmatrix} \rho u \\ \rho u^2 + p - \tau_{xx} \\ \rho uv - \tau_{xy} \\ (E_t + p)u - u\tau_{xx} - v\tau_{xy} + q_x \end{Bmatrix}$$

$$F = \begin{Bmatrix} \rho v \\ \rho uv - \tau_{xy} \\ \rho v^2 + p - \tau_{xy} \\ (E_t + p)v - u\tau_{xy} - v\tau_{yy} + q_y \end{Bmatrix}$$

Apart from these, the additional equations used, that forms complete Navier-Stokes equations are given below.

$$\tau_{xy} = \tau_{yx} = \mu \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right)$$

$$\tau_{xx} = \lambda \left(\nabla . \vec{V} \right) + 2\mu \frac{\partial u}{\partial x}$$

$$\tau_{yy} = \lambda \left(\nabla . \vec{V} \right) + 2\mu \frac{\partial v}{\partial y}$$

$$q_x = -k \frac{\partial T}{\partial x}$$

$$q_y = -k \frac{\partial T}{\partial y}$$

$$p = \rho RT$$

$$e = C_v T$$

$$\mu = \mu_0 \left(\frac{T}{T_0} \right)^{3/2} \frac{T_0 + 110}{T + 110}$$

$$k = \frac{\mu C_p}{Pr}$$

The equation used in computing the timestep using Courant-Friedrich-Lewy condition is given below. The following equation is used to compute timestep size for all internal grid points and the timestep for computation is chosen to be the lowest of all multiplied with Courant number $K = 0.6$.

$$(\Delta t_{CFL})_{i,j} = \left[\frac{|u_{i,j}|}{\Delta x} + \frac{|v_{i,j}|}{\Delta y} + \sqrt{\gamma R T_{i,j}} \sqrt{\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2}} + 2\nu_{i,j} \left(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} \right) \right]^{-1}$$

$$\Delta t = K \times min \left(\Delta t_{CFL} \right)$$

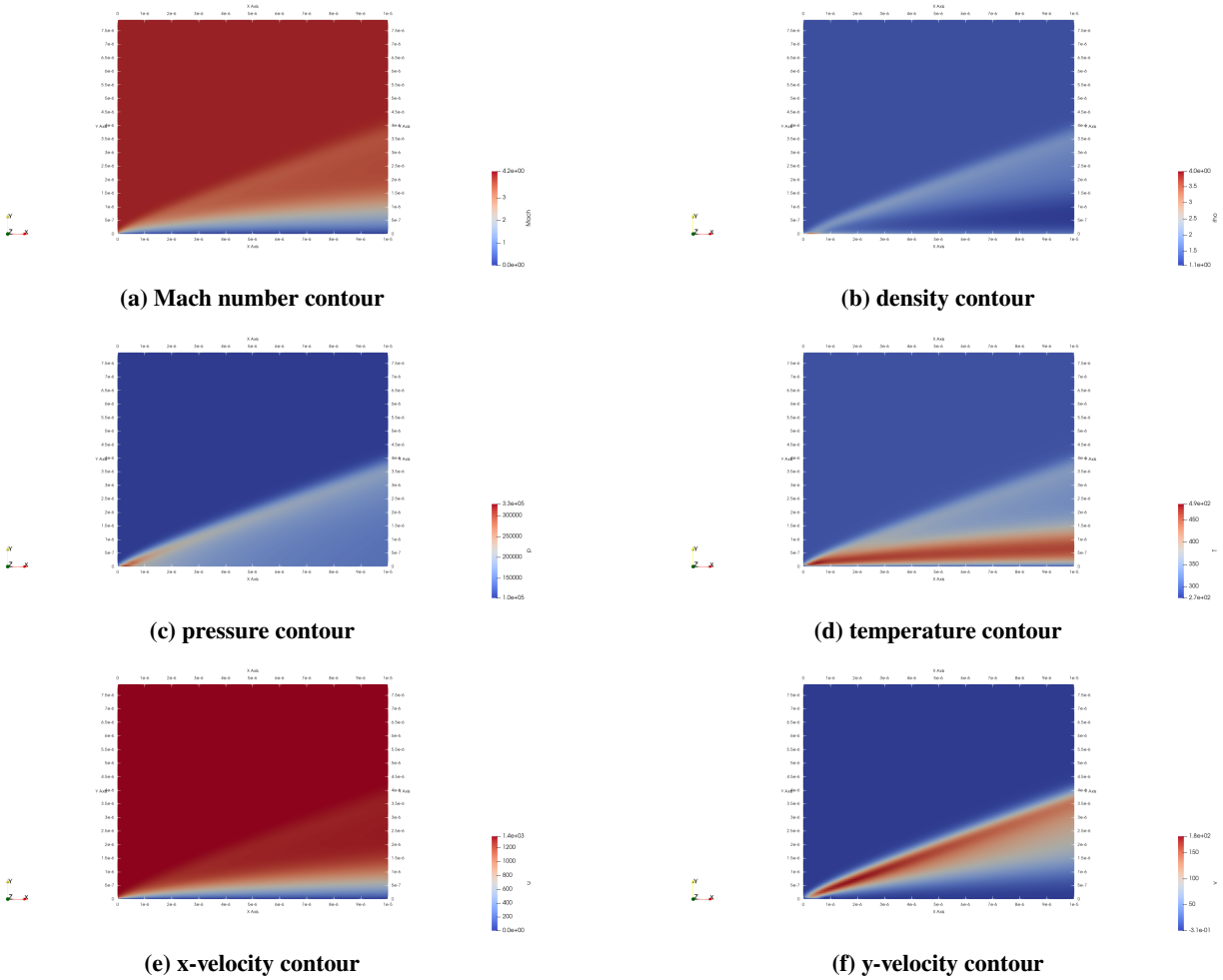
The algorithm followed for the computation is given below.

1) The program is setup with required variables. The flow field variables such as temperature and pressure were initialized to free-stream values.
2) The boundary conditions are applied, as indicated in the Figure 1.
3) The conservative variables, U's, were computed from the primitive flow field variables. And the computation timestep is computed based on the flow field variables.
4) The shear stresses and heat flux were computed using the derivatives of velocity that were computed from the current velocity fields.
5) Then the conservative variables, E's and F's were computed from the primitive, shear stresses and heat flux variables.

6) For the predictor step, the derivatives of E and F were computed using forward differencing and U derivatives were computed acoordingly.
7) Then the predicted U values were computed and used to compute predicted primitive variables, then boundary conditions were applied and then the values were used to compute E's and Fs for the corrector step.
8) The U derivatives of corrector step were computed from E and F derivatives and then the average of U derivatives from predictor and corrector steps were used to compute U values of next timestep.
9) The primitive variables for next timestep is computed from the new U values and the boundary conditions were imposed, then the process is repeated from step 3 till convergence.
10) Convergence is confirmed when the change in density values between each timestep is to be less than 1e-8.

III. Computation results

The Fortran code is made in a modular way, such that any further modifications and enhancements can be easily implemented. The converged solution is written as a csv file from the Fortran code and the csv file is loaded into ParaView for visualization. The contours obtained for the present case is shown in Figure 2, this case is with constant wall temperature set to free-stream temperature value.



(a) Mach number contour

(b) density contour

(c) pressure contour

(d) temperature contour

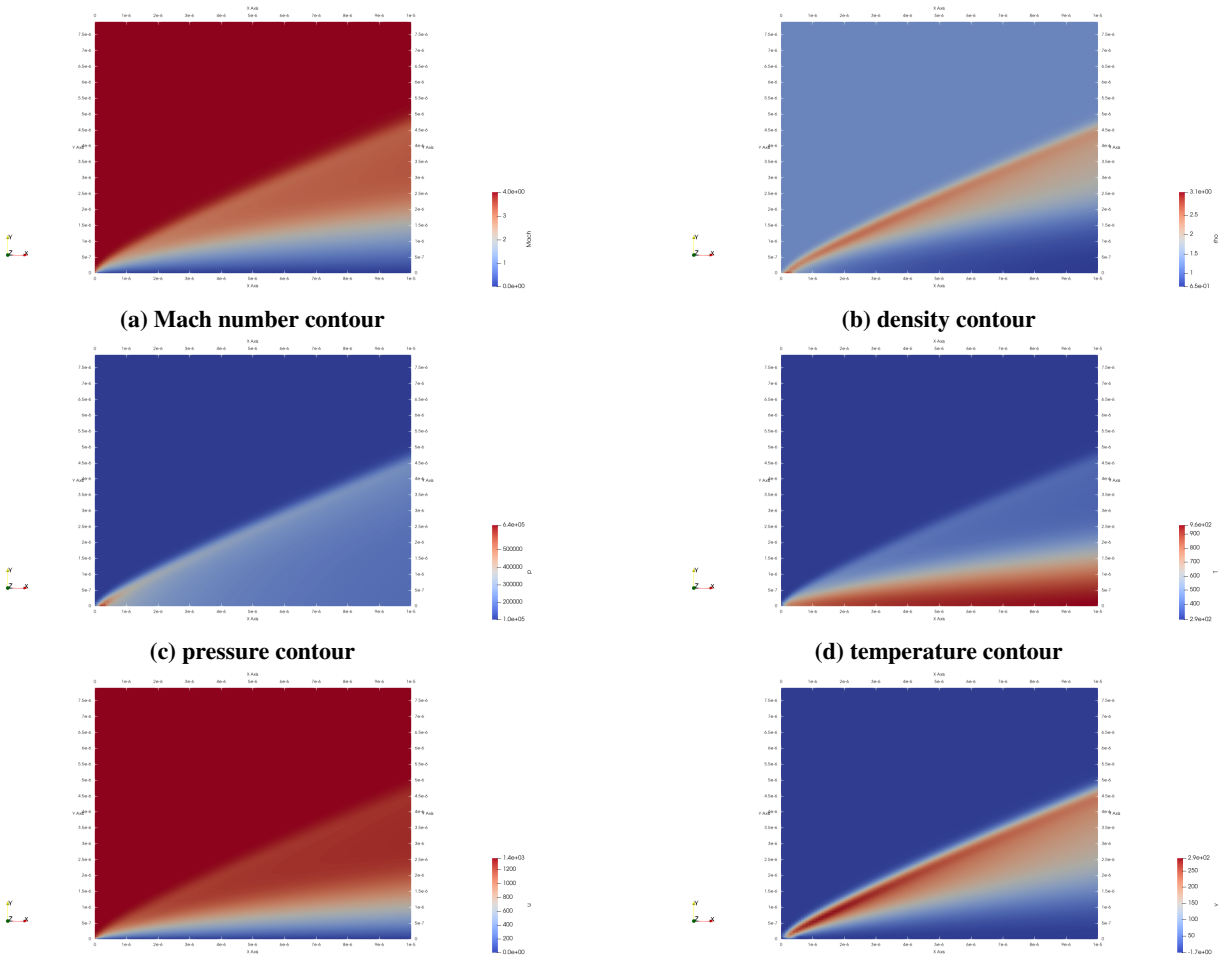
(e) x-velocity contour

(f) y-velocity contour

Fig. 2 Computation results of case with constant wall temperature, set to free-stream temperature

Further, another case with adiabatic wall temperature was computed, those results are given in Figure 3

And, another case with flow entering with an angle of -10 degrees with x-axis i.e. flowing downwards with 10 degrees with the horizontal, is made by modifying boundary conditions to simulate the supersonic flow over a wedge



(a) Mach number contour

(b) density contour

(c) pressure contour

(d) temperature contour

(e) x-velocity contour

(f) y-velocity contour

Fig. 3 Computation results of case with adiabatic wall temperature

inclined at 10 degrees with flow direction. The results obtained in this case is validated by comparing the average downstream Mach number with the theorectical value. The results are given in Figure 4.

Clear contours for all the cases were present with the code in the github repository.

IV. list of numerical experiments

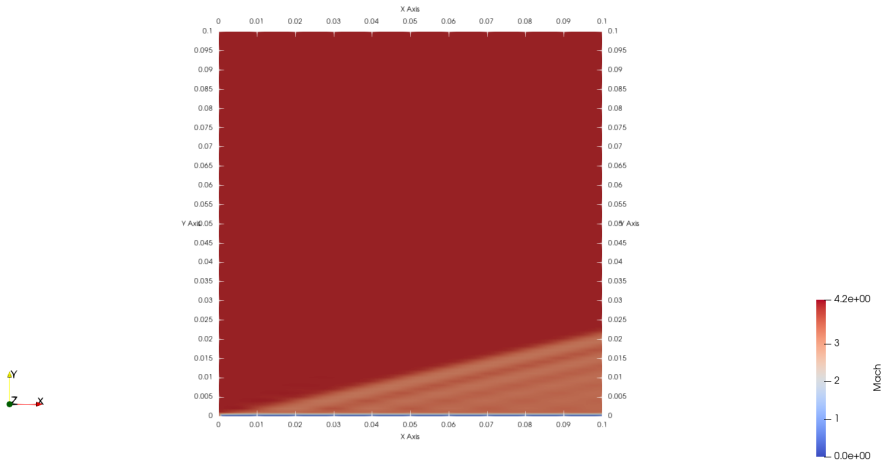
Following is the list of numerical experiments made with the versions of this solver

1) laminar supersonic flow over flat plate (version 1 solver)
2) laminar subsonic boundary layer (version 2 solver)
3) laminar subsonic channel flow (v2 solver)
4) laminar subsonic lid-driven cavity flow (v2 solver)

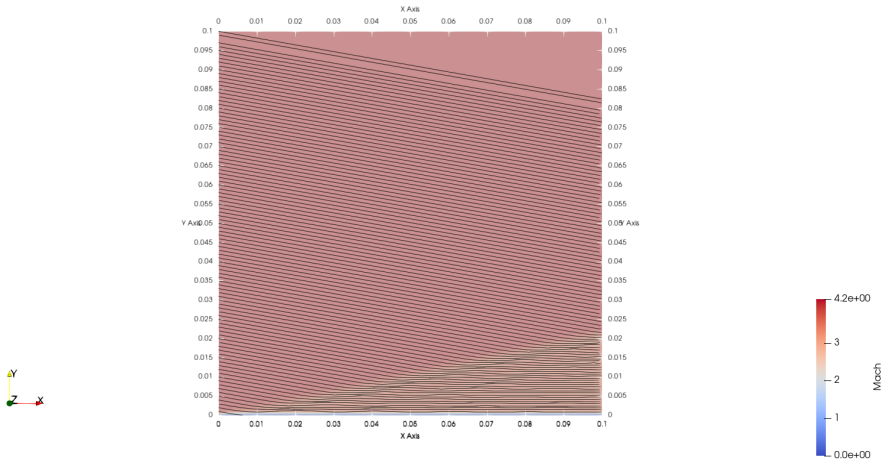
V. Conclusion

The laminar viscous supersonic flow over flat plate were numerically computed using Finite difference method and full Navier-Stokes equations. The Fortran code is made modular so that any further modifications and enhancements can be implemented. The following enhancements are made to the solver's current version 2

- simulation resume capability
- chaning matrix multiplications to loops for speed enhancement (solver v2)



(a) Mach number contour



(b) streamline pattern showing the flow direction

Fig. 4 Computation results of case simulating oblique shock

The Fortran code, along with this documentation can be found in the Github repository at `https://github.com/Ramkumar47/ComputationalCodes/05_supersonicFlow_over_flatPlate/`

****