Potential Flow over Ellipse body

Ramkumar

August 21, 2022

1 Problem Definition

In this work, the potential flow over ellipse body is simulated using potential flow theory. The code was made as a custom application in OpenFOAM v2206 and the postprocessing is done using ParaView. The main aim of this work is to introduce the OpenFOAM programming powers to the users.

2 Governing equations

In this work, the potential flow elements containing a uniform flow, a source and a sink positioned appropriately to form an ellipse body inside flowfield. The main equation that is used to compute streamfunction ψ is given in Equation (1).

$$\psi = V_\infty \cdot r \cdot sin(\theta) + \frac{\Lambda}{2\pi} \left(\theta_{sink} - \theta_{source}\right) \tag{1}$$

where,

V_∞	-	free stream velocity magnitude
θ	-	angle w.r.t. global origin
θ_{sink}	-	angle w.r.t. sink center
θ_{source}	-	angle w.r.t. source center
r	-	radial distance w.r.t. global center
Λ	-	source and sink's strength

The velocity components in cartesian coordinates were calculated by taking gradient of ψ field, as shown in Equations (2) and (3).

$$U_x = \frac{\partial \psi}{\partial y} \tag{2}$$

$$U_y = -\frac{\partial \psi}{\partial x} \tag{3}$$

A rectangular mesh was made using *blockMesh* utility in OpenFOAM and the cell centers and boundary face centers in the mesh were used as the computation points for streamfunction ϕ and other variables. The radial distance r and azimuthal distance θ are calculated using the Equations 4 and 5, respectively.

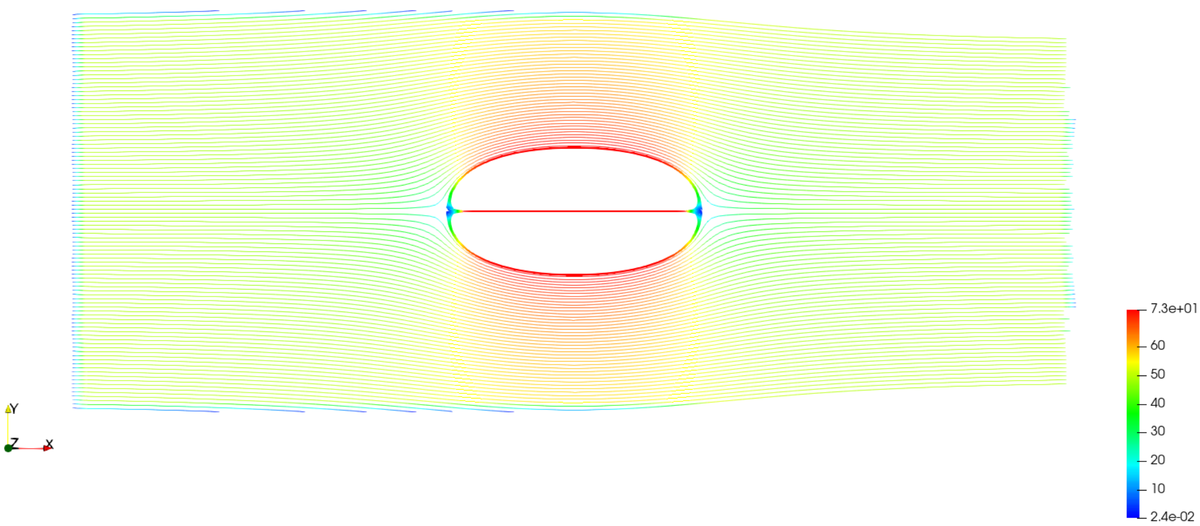

Figure 1: streamlines contour of semi-infinite body

$$r = \sqrt{\Delta x^2 + \Delta y^2} \tag{4}$$
$$\theta = arctan2(\Delta y, \Delta x) \tag{5}$$

3 Computation Methodology

The steps followed in OpenFOAM to generate the semi-infinite body are given below.

1. a new custom application was made in OpenFOAM v2206 and named as *PotentialFlow_ellipse* using command *foamNewApp appName*.

2. totally new 1 volScalarField and 1 volVectorField were created to store computed ψ and U using the equations mentioned in the Section 2.

3. a loop over cells is created using which the field values for each cell is computed and a similar loop is created to go over boundary faces, then at the end of 2^{nd} loop the solution fields will be writen to the *0* folder in the case directory.

The result obtained using the OpenFOAM code is shown as a contour with streamlines that were made using *ParaView*, in Figure 1.

4 Instructions

The instruction to generate/execute the files present in this work is given below.

1. copy the contents of this entire root folder named *02_PotentialFlowOverEllipse* to a new directory.

2. go into the subfolder named *03_OpenFOAMCode* which contains another sub-subfolder named *PotentialFlow_ellipse*, go into that using the terminal which is enabled with OpenFOAM environment.

3. in the terminal, type *wclean* and press enter, to clean any previous compilation files. Then type *wmake* and press enter, this will compile the application.

4. after compilation, cd to the directory named *02_testCase* and execute the command named *PotentialFlow_ellipse*. this should compute the field values and store them in the *0* folder.

5. after computation, use *ParaView* for visualization of results.