Assignment 1: Advanced computational fluid dynamics (ME670)-2024

Department of Mechanical Engineering, Indian Institute of Technology Guwahati

- Please provide: (i) the grid detail, (ii) the discretized equations detail, (iii) the boundary condition implementation detail, (iv) a well-documented code, (v) the required output (plots/any other such means).
- Items (i), (ii), and (iii) above should be written out/typed on a separate sheet and attached before items (iv) and (v).

Consider the lid-drive cavity model problem shown in fig. 1. The square cavity is formed by three stationary walls (sides and bottom) and one moving (top) wall of length L. The cavity is very long along the z-direction. It is filled with a Newtonian fluid of density ρ and viscosity μ. The top lid/wall moves with a velocity $(u, v) = (U, 0)$ m/s. The flow can be assumed two-dimensional, incompressible, steady and isothermal. Solve the non-dimensional steady Navier-Stokes equations using the finite volume method on a staggered grid and SIMPLE scheme.

Use a 129×129 uniform and Cartesian finite volume grid. Use the Hybrid differencing scheme for solving the momentum equations. For convergence, use the $\|L_\infty\| < 10^{-5}$ criterion. For the solution of the discretized equations, you can use the point Gauss-Seidel (GS) or ADI method.

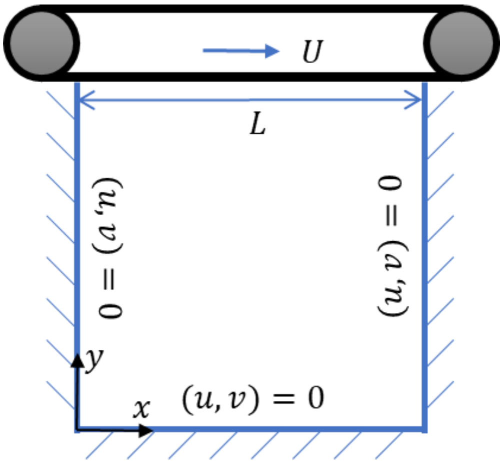

Figure1: A schematic of the lid-drive cavity problem

For Re $(= \rho U/L)$ = 100, 400 and 1000

1. Show contours for velocity magnitude and vorticity (ω). Also show streamline pattern by plotting the contours of stream function (ψ).
2. Plot the x-velocity profile at $x = L/2$, and x-velocity profile at $y = L/2$. Compare your results by plotting the values from Tables I and II in the paper by Ghia et al.
3. Compare the following results for the primary vortex given in Table V of Ghia et al. with your results: $\psi_{\min}$, location (x, y) of $\psi_{\min}$ and the value of ω at the location of $\psi_{\min}$.

Coding guidelines/tips:

- You can use C/C++ or Fortran (except Fortran 77) programming languages for writing your codes. Both GNU and intel compilers can be used free of cost.
- Pay attention to the row/column major storage schemes of C and Fortran. Discontinuous memory excess significantly slows down a CFD code.
- It is advisable to declare appropriate variables as three-dimensional arrays so that later one can add three-dimensionality to the code with ease.
- All major operations such as calculation of coefficients, one SIMPLE iteration, application of boundary conditions, point GS/ADI, file writing, etc. should be done through the use of functions/subroutines.
- Fortran users should learn the use of *Modules* and *kind* specifiers.
- User inputs such as grid size, Reynolds number, convergence criteria, boundary conditions, etc. should be read from an ASCI formatted text file called "input.dat".
- For plotting you can use Tecplot, Matplotlib (Python), or Gnuplot. Matplotlib is my personal preference for line plots.
- Use compiler optimization flags like -O2.
- Variable names should be self-explanatory and there should be enough comments to describe the working and flow of your code. There is an open-source tool called *doxygen* for code documentation (learn yourself if interested).
- There are IDEs such as VScode, eclipse, vim, etc. that offer advanced editing features to help you write codes with ease. It is okay to use normal text editors if this seems too much to you.
- Use of Linux-based operating systems, such as Ubuntu, is encouraged. You can learn shell scripting for automating repetitive tasks.

References:

Ghia, U. K. N. G., Ghia, K. N., & Shin, C. T. (1982). High-Re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method. Journal of computational physics, 48(3), 387-411.

Bruneau, C. H., & Saad, M. (2006). The 2D lid-driven cavity problem revisited. Computers & fluids, 35(3), 326-348.