- 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).

Using finite difference/finite volume method, write a code for solving the following unsteady NaiverStokes equations (in non-dimensional form) governing the incompressible flow of a Newtonian fluid

$$\frac{\partial u_i}{\partial x_i} = 0$$

$$\frac{\partial u_i}{\partial t} + \frac{\partial}{\partial x_j}(u_i u_j) = -\frac{\partial p}{\partial x_i} + \frac{1}{Re}\frac{\partial}{\partial x_j}\left(\frac{\partial u_i}{\partial x_j}\right)$$

Assume a non-uniform structured Cartesian grid where the unknowns are stored at cell centers. Use Ghost-cell approach for applying boundary conditions. Use P1 fractional-step method for solving NS equations and CDS scheme for all spatial derivatives. The mesh spacing changes with a constant ratio. Using your code, solve the following two problems:

Problem 1: Driven Cavity flow

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, and isothermal. Use a 128×128 uniform and Cartesian grid. For convergence, use the $\|L_\infty\| < 10^{-5}$ criterion.
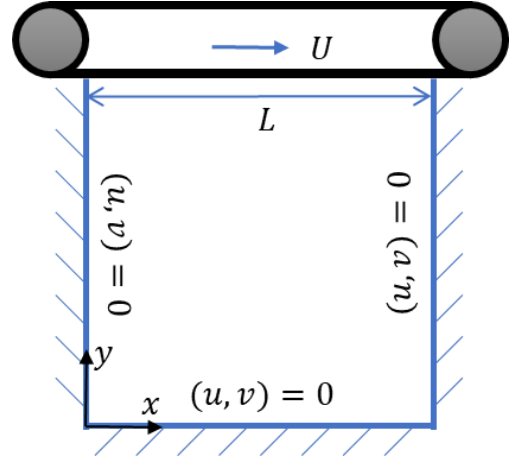

Figure 1: A schematic of the lid-drive cavity problem

For Re ($= UL/v$) = 100, 400 and 1000, at steady-state,

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 y-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 [1].
3. Compare the following results for the primary vortex given in Table V of Ghia et al. [1] with your results: ψ_{min}, location (x, y) of ψ_{min} and the value of ω at the location of ψ_{min}.

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:

1. 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.
2. A. Sohankar, C. Norberg, and L. Davidson, "Numerical simulation of unsteady flow around a square two-dimensional cylinder," Twelfth Australasian Fluid Mechanics Conference, 1995.