CHALMERS UNIVERSITY OF TECHNOLOGY
Fluid Dynamics, M2
412 96 Göteborg

MTF072 Computational Fluid Dynamics

H. Nilsson, D. Ghosh, K. Konstantinidis November 24, 2022

Task 3

In this task you will develop a 2D Navier-Stokes segregated solver with the SIMPLE algorithm for incompressible, steady state, laminar flow with constant kinematic viscosity. **As before, this task should be carried out using either Python in groups of up to two students**. You should compute the flow in a square ($L = H = 1$) lid-driven cavity in a case with $Re = U_{wall}L/\nu = 1000$, by setting $U_{wall} = L = \rho = 1$ and $\nu = 1/1000$. A lid-driven cavity case can be thought of as a box with fluid inside where one of the walls is moving (in this case the top wall), as depicted in Fig. 1.

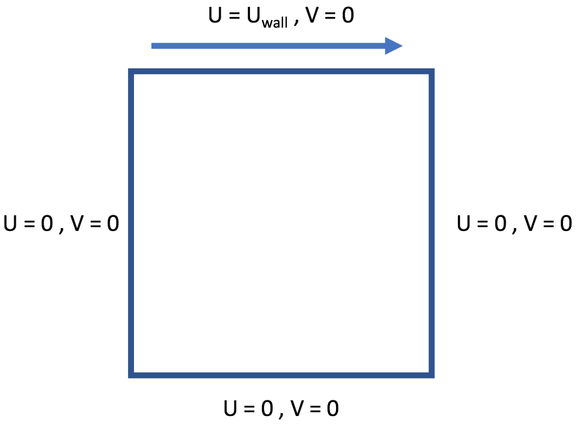

Figure 1: *Schematic of the lid-driven cavity problem.*

The equations that govern the flow behaviour are, the 2D momentum conservation equations for incompressible, steady state, laminar flow with constant kinematic viscosity together with the continuity equation. These can be written as,

$$\frac{\partial U}{\partial x} + \frac{\partial V}{\partial y} = 0 \tag{1}$$

$$U\frac{\partial U}{\partial x} + V\frac{\partial U}{\partial y} = -\frac{1}{\rho}\frac{\partial P}{\partial x} + \nu\frac{\partial^2 U}{\partial x^2} + \nu\frac{\partial^2 U}{\partial y^2} \tag{2}$$

$$U\frac{\partial V}{\partial x} + V\frac{\partial V}{\partial y} \;=\; -\frac{1}{\rho}\frac{\partial P}{\partial y} + \nu\frac{\partial^2 V}{\partial x^2} + \nu\frac{\partial^2 V}{\partial y^2} \tag{3}$$

where Eqs. 1, 2 and 3 represent the continuity, U-momentum and V-momentum conservation equations respectively. Note that due to the incompressibility assumption, density has been taken out of the derivatives of the left hand side and brought to the right hand side in the two momentum equations.

The algebraic equation system should be solved using:

- Gauss-Seidel

How to proceed

- In order to help you with the task, a template has been prepared in Python (`task_3_template.py`), available on Canvas. Instructions on how to approach this problem can be found in this description and in the comments of the template. **Make sure to use the template and the same nomenclature** for the variables as given in it, so that getting help from the assistants becomes easier.
- **Regarding the schemes, use the first-order upwind (FOU) convection scheme**.
- An equidistant grid is provided to you.
- Solve for your velocity and pressure fields using the Gauss-Seidel method. You should mostly use `for` (or `while`) loops in your script (avoid solving with commands like *"backslash"*). Notice that we solve the equation system in a co-located grid.
- Implicit under-relaxation has to be introduced to help with convergence.
- Rhie-Chow interpolation must be used in order to avoid checkerboard-like pressure solutions, a common issue encountered when using co-located grids to solve the coupled system of Eqs. 1, 2 and 3.
- The boundary condition at all boundaries for the pressure correction is homogeneous Neumann in their corresponding wall normal directions.
- **Note that the pressure correction equation can be re-solved for each Gauss-Seidel iteration of the momentum equations.** This will facilitate convergence. It will only be needed a few iterations to make the pressure correction converge (maybe 10 or few more).
- Since homogeneous Neumann boundary conditions are applied at all boundaries for the pressure correction equation, the exact pressure correction level is not defined. Hence, choose a certain node (e.g. left-bottom cell) and set its value as a reference pressure by setting it to zero in each iteration. The pressure correction field needs to be adjusted accordingly.

- Implement the residual criteria as in Eq. 4 for the U and V momentum conservation as well as continuity. The residual of the continuity equation is the sum of the absolute net flux in each node (which is also the source term in the pressure correction equation).
- Compare the results that you obtain with the provided data (**download the** `data_FOU_CD.txt` **file from Canvas and place it in the same directory as the** *.py* **file**) at a line located at $x = 0.5$. Once you have correctly computed the U and V matrices, use the plotting function provided in the template to generate a comparison plot.
- **In order to be able to make a fair comparison with the provided data, mI and mJ (number of mesh points in both directions) should be kept equal to** 11.
- A second data file is given with the Hybrid scheme implemented. You can give a comparison of your results with these data as well, or you can even adjust your coefficients to the hybrid scheme and compare (optional).

Convergence

It is very important to verify that a converged solution has been obtained. At each iteration compute the residual for U and V momentum equations as

$$\varepsilon = \frac{1}{F}\left(\sum_{\text{allnodes}} |a_P^u U_P - (a_E^u U_E + a_W^u U_W + a_N^u U_N + a_S^u U_S + Su^u)|\right) \tag{4}$$

where F can be set to equal to 1 as it can be a representative value of the problem in order to make the residuals non-dimensional (the residual without normalization has units of m^3/s^2, hence it makes sense to set $F = U_{wall}^2 * L$ in order to make the normalized residuals dimensionless as it should be). The residual for continuity equation is the magnitude of the net flux in each node. The solution is considered as converged when $\varepsilon < 10^{-4}$. You can of course change this threshold to see if any differences are visible.

Presentation of the work

The work should be presented in form of a short report (max 10 pages without taking into account the appended code. **This is a strict limit**). The code **must** be included as an appendix at the end of the report. Try to discuss the results from a physical and numerical point of view. Present relevant results for example as contour plots of the computed variables etc. to support your findings. The report must include the following parts:

- Description on the discretization process.
- U, V and P contour fields.
- In order to illustrate the re-circulation zones within the domain, plot the velocity vector field and relevant contours.

- The comparison with the provided data. How good is your code?
- Verify and plot your residuals against the number of iterations.