

TVD_Fluid_Simulator

I. General Discretized Equations:

To derive the 2D incompressible momentum equation written in the non-dimensional form, it might as well introduce the dimensionless term $u^* = \frac{u}{U}$, $x^* = \frac{x}{L}$, $t^* = \frac{t}{U/L}$, $p^* = \frac{p}{\rho U^2}$ (where U is the lid velocity, L is the square cavity's side length, and ρ the constant) into the original form:

$$\frac{\partial u}{\partial t} + u \cdot \nabla u = -\frac{1}{\rho} \nabla p + v \nabla^2 u,$$

thus transferred as:

$$\frac{\partial u^*}{\partial t^*} \frac{U^2}{L} + u^* \cdot \nabla u^* \frac{U^2}{L} = -\frac{1}{\rho} \nabla p^* \frac{\rho U^2}{L} + v \nabla^2 u^* \frac{U}{L^2},$$

equals to:

$$\frac{\partial u^*}{\partial t^*} + u^* \cdot \nabla u^* = -\nabla p^* + \frac{v}{UL} \nabla^2 u^*.$$

Thus the Reynolds number could be defined as $Re = \frac{UL}{v}$, and for detailed simulation here could assume the kinematic viscosity constant as $1 \times 10^{-3} \text{ m}^2/\text{s}$ with given length L as 1, the Re number varying from 1 to 1000 corresponds to the velocity change from 1 mm/s to 1m/s.

For the initial and boundary conditions, boundary velocities are already specified in the question statement, and here assumes the initial flow velocity is 0 throughout the interior domain to reach a steady state. For the pressure, set the reference value as 1 atm, and assume the initial pressure field is 0 for all inner space. As for its boundary, set all four sides the zero-gradient type in the normal direction due to the laminar boundary layer analysis on the wall.

For the numerical schemes used in the discretized momentum equations, here adapts fully implicit scheme for the unsteady term, second order accuracy central difference scheme for the pressure gradient and velocity gradient in diffusion term and third order (where monotonic, r as defined below, > 0) accuracy TVD scheme UMIST for the convection term.

The UMIST scheme is first written for one direction as

$$a_P U_P = a_W U_W + a_E U_E + S_u,$$

where

$$\begin{aligned} a_P &= a_W + a_E + F_e - F_w + a_P^0, \\ a_w &= D_w + \max(F_w, 0), \\ a_E &= D_e + \max(-F_e, 0), \end{aligned}$$

$$S_u = \frac{1}{2} F_e [(1 - \alpha_e) \Psi(r_e^-) - \alpha_e \cdot \Psi(r_e^+)] (U_E - U_P)$$

$$+\frac{1}{2}F_w[\alpha_w \cdot \Psi(r_w^+) - (1 - \alpha_w)\Psi(r_w^-)](U_P - U_W) \\ + a_p^0 U_P^0 + \Delta P,$$

$$r_e^+ = \frac{U_P - U_W}{U_E - U_P},$$

$$r_e^- = \frac{U_{EE} - U_E}{U_E - U_P},$$

$$r_w^+ = \frac{U_W - U_{WW}}{U_P - U_W},$$

$$r_w^- = \frac{U_E - U_P}{U_P - U_W},$$

$$\Psi(r) = \max \left[0, \min \left(2r, \frac{1+3r}{4}, \frac{3+r}{4}, 2 \right) \right].$$

Set $dx = dy = \Delta L$,

$$F_w = \frac{U_P + U_W}{2},$$

$$F_e = \frac{U_P + U_E}{2},$$

$$D_w = \frac{1}{Re\Delta L},$$

$$D_e = \frac{1}{Re\Delta L},$$

$$a_p^0 = \frac{\Delta L}{\Delta t},$$

$$\Delta P = P_w - P_e.$$

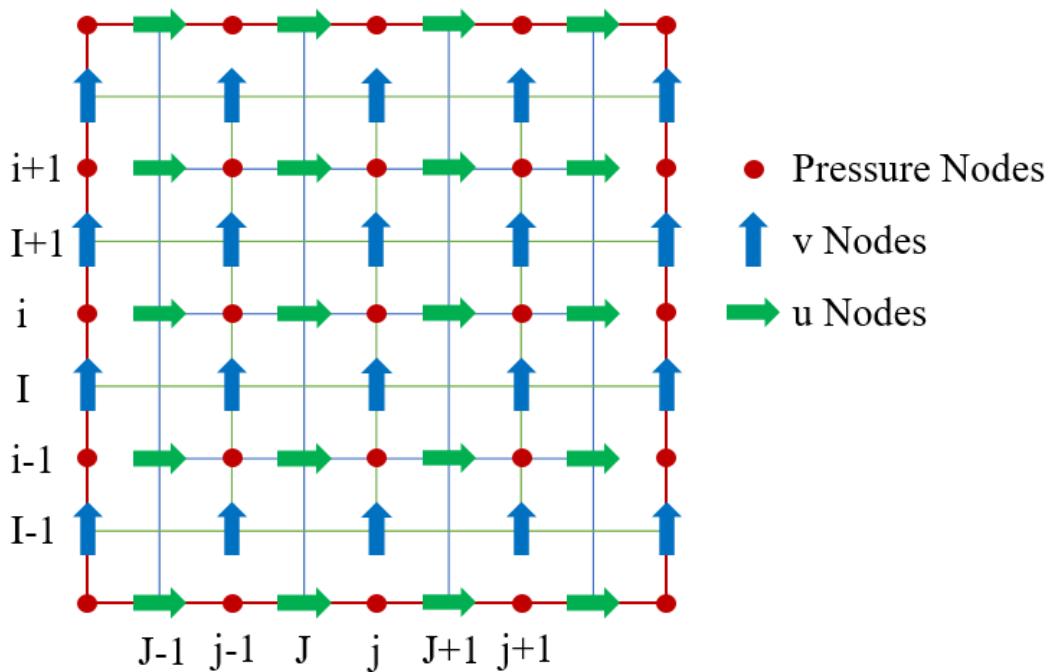


Figure 1. Illustration of 2-D staggered grids of u , v , and p .

Therefore, constructing the staggered grids as above shown in Fig. 1, denote the pressure cell with subscripts I, J as well as the velocity cell with subscripts i, j in a backward manner to that of pressure, and the 2D discretized momentum equation would be shaped as following.

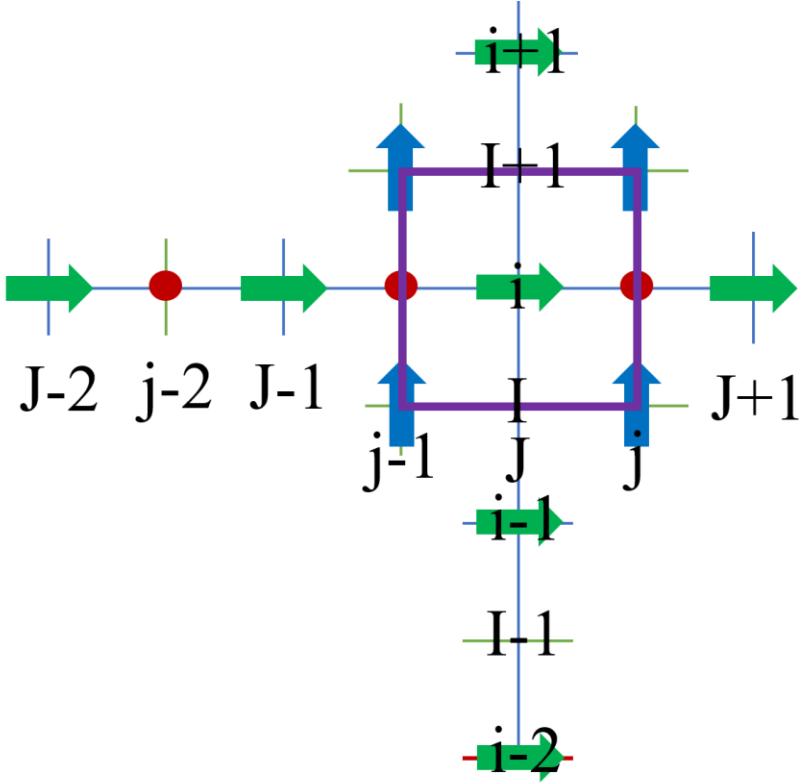


Figure 2. Illustration of U cell.

i. u momentum equation:

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot \vec{\mathbf{U}}\mathbf{u} = -\nabla \cdot (\mathbf{p}, 0) + \frac{1}{Re} \nabla \cdot (\nabla \mathbf{u}).$$

The discretized form would be:

$$\begin{aligned} & \Delta L^2 \frac{\mathbf{u}_{i,J} - \mathbf{u}_{i,J}^0}{\Delta t} + (F_{i,j}\mathbf{u}_{i,j} - F_{i,j-1}\mathbf{u}_{i,j-1} + F_{I+1,J}\mathbf{u}_{I+1,J} - F_{I,J}\mathbf{u}_{I,J})\Delta L \\ &= -(\mathbf{p}_{i,j} - \mathbf{p}_{i,j-1})\Delta L \\ &+ \frac{1}{Re} \left(\frac{\mathbf{u}_{i,J+1} - \mathbf{u}_{i,J}}{\Delta L} + \frac{\mathbf{u}_{i+1,J} - \mathbf{u}_{i,J}}{\Delta L} - \frac{\mathbf{u}_{i,J} - \mathbf{u}_{i,J-1}}{\Delta L} - \frac{\mathbf{u}_{i,J} - \mathbf{u}_{i-1,J}}{\Delta L} \right) \Delta L, \end{aligned}$$

And for above four u's value on the boundaries, it is evaluated exactly as aforementioned UMIST scheme for both directions.

It is then in the from:

$$a_{i,J}\mathbf{u}_{i,J} = a_{i-1,J}\mathbf{u}_{i-1,J} + a_{i+1,J}\mathbf{u}_{i+1,J} + a_{i,J-1}\mathbf{u}_{i,J-1} + a_{i,J+1}\mathbf{u}_{i,J+1} + S_{i,J},$$

where

$$a_{i,J} = a_{i-1,J} + a_{i+1,J} + a_{i,J-1} + a_{i,J+1} + F_{i,j} - F_{i,j-1} + F_{I+1,J} - F_{I,J} + a_p^0,$$

$$a_{i-1,J} = \frac{1}{Re\Delta L} + \max(F_{I,J}, 0),$$

$$a_{i+1,J} = \frac{1}{Re\Delta L} + \max(-F_{I+1,J}, 0),$$

$$a_{i,J-1} = \frac{1}{Re\Delta L} + \max(F_{i,j-1}, 0),$$

$$a_{i,J+1} = \frac{1}{Re\Delta L} + \max(-F_{i,j}, 0),$$

$$\begin{aligned} S_{i,J} &= \frac{1}{2} F_{I+1,J} [(1 - \alpha_{I+1,J}) \Psi(r_{I+1,J}^-) - \alpha_{I+1,J} \cdot \Psi(r_{I+1,J}^+)] (u_{i+1,J} - u_{i,J}) \\ &\quad + \frac{1}{2} F_{I,J} [\alpha_{I,J} \cdot \Psi(r_{I,J}^+) - (1 - \alpha_{I,J}) \Psi(r_{I,J}^-)] (u_{i,J} - u_{i-1,J}) \\ &\quad + \frac{1}{2} F_{i,j} [(1 - \alpha_{i,j}) \Psi(r_{i,j}^-) - \alpha_{i,j} \cdot \Psi(r_{i,j}^+)] (u_{i,J+1} - u_{i,J}) \\ &\quad + \frac{1}{2} F_{i,j-1} [\alpha_{i,j-1} \cdot \Psi(r_{i,j-1}^+) - (1 - \alpha_{i,j-1}) \Psi(r_{i,j-1}^-)] (u_{i,J} - u_{i,J-1}) \\ &\quad + a_p^0 u_{i,J}^0 + \Delta P, \end{aligned}$$

$$r_{I+1,J}^+ = \frac{u_{i-1,J} - u_{i,J}}{u_{i,J} - u_{i+1,J}},$$

$$r_{I+1,J}^- = \frac{u_{i+2,J} - u_{i+1,J}}{u_{i+1,J} - u_{i,J}},$$

$$r_{I,J}^+ = \frac{u_{i-2,J} - u_{i-1,J}}{u_{i-1,J} - u_{i,J}},$$

$$r_{I,J}^- = \frac{u_{i+1,J} - u_{i,J}}{u_{i,J} - u_{i-1,J}},$$

$$r_{i,j}^+ = \frac{u_{i,J-1} - u_{i,J}}{u_{i,J} - u_{i,J+1}},$$

$$r_{i,j}^- = \frac{u_{i,J+2} - u_{i,J+1}}{u_{i,J+1} - u_{i,J}},$$

$$r_{i,j-1}^+ = \frac{u_{i,J-2} - u_{i,J-1}}{u_{i,J-1} - u_{i,J}},$$

$$r_{i,j-1}^- = \frac{u_{i,J+1} - u_{i,J}}{u_{i,J} - u_{i,J-1}}.$$

$$\Psi(r) = \max \left[0, \min \left(2r, \frac{1+3r}{4}, \frac{3+r}{4}, 2 \right) \right],$$

$\alpha_x = 1$ if $F_x > 0$ and $\alpha_x = 0$ if $F_x < 0$, (no need to consider α_x if $F_x = 0$),

$$F_{i,j} = \frac{u_{i,J} + u_{i,J+1}}{2},$$

$$F_{i,j-1} = \frac{u_{i,J-1} + u_{i,J}}{2},$$

$$F_{I+1,J} = \frac{v_{I+1,j-1} + v_{I+1,j}}{2},$$

$$F_{I,J} = \frac{v_{I,j-1} + v_{I,j}}{2},$$

$$a_p^0 = \frac{\Delta L}{\Delta t},$$

$$\Delta P = p_{i,j-1} - p_{i,j}.$$

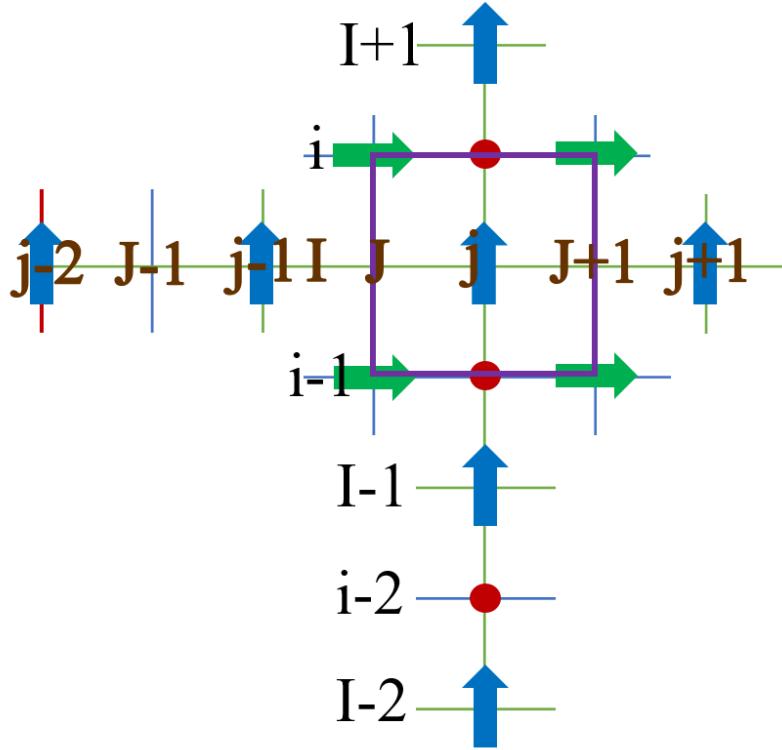


Figure 2. Illustration of V cell.

ii. v momentum equation:

Similarly,

$$\frac{\partial v}{\partial t} + \nabla \cdot \vec{U}v = -\nabla \cdot (0, p) + \frac{1}{Re} \nabla \cdot (\nabla v).$$

The discretized form would be:

$$\begin{aligned} & \Delta L^2 \frac{v_{I,j} - v_{I,j}^0}{\Delta t} + (F_{I,j+1} v_{I,j+1} - F_{I,j} v_{I,j} + F_{i,j} v_{i,j} - F_{i-1,j} v_{i-1,j}) \Delta L \\ &= -(p_{i,j} - p_{i-1,j}) \Delta L \\ &+ \frac{1}{Re} \left(\frac{v_{I,j+1} - v_{I,j}}{\Delta L} + \frac{v_{I+1,j} - v_{I,j}}{\Delta L} - \frac{v_{I,j} - v_{I,j-1}}{\Delta L} - \frac{v_{I,j} - v_{I-1,j}}{\Delta L} \right) \Delta L. \end{aligned}$$

And further implementing the UMIST scheme, it would take the form:

$$a_{I,j} v_{I,j} = a_{I-1,j} v_{I-1,j} + a_{I+1,j} v_{I+1,j} + a_{I,j-1} v_{I,j-1} + a_{I,j+1} v_{I,j+1} + S_{I,j},$$

where

$$a_{I,j} = a_{I-1,j} + a_{I+1,j} + a_{I,j-1} + a_{I,j+1} + F_{I,J+1} - F_{I,J} + F_{i,j} - F_{i-1,j} + a_p^0,$$

$$a_{I-1,j} = \frac{1}{Re\Delta L} + \max(F_{i-1,j}, 0),$$

$$a_{I+1,j} = \frac{1}{Re\Delta L} + \max(-F_{i,j}, 0),$$

$$a_{I,j-1} = \frac{1}{Re\Delta L} + \max(F_{I,J}, 0),$$

$$a_{I,j+1} = \frac{1}{Re\Delta L} + \max(-F_{I,J+1}, 0),$$

$$S_{I,j} = \frac{1}{2} F_{i,j} [(1 - \alpha_{i,j}) \Psi(r_{i,j}^-) - \alpha_{i,j} \cdot \Psi(r_{i,j}^+)] (v_{I+1,j} - v_{I,j})$$

$$+ \frac{1}{2} F_{i-1,j} [\alpha_{i-1,j} \cdot \Psi(r_{i-1,j}^+) - (1 - \alpha_{i-1,j}) \Psi(r_{i-1,j}^-)] (v_{I,j} - v_{I-1,j})$$

$$+ \frac{1}{2} F_{I,J+1} [(1 - \alpha_{I,J+1}) \Psi(r_{I,J+1}^-) - \alpha_{I,J+1} \cdot \Psi(r_{I,J+1}^+)] (v_{I,j+1} - v_{I,j})$$

$$+ \frac{1}{2} F_{I,J} [\alpha_{I,J} \cdot \Psi(r_{I,J}^+) - (1 - \alpha_{I,J}) \Psi(r_{I,J}^-)] (v_{I,j} - v_{I,j-1})$$

$$+ a_p^0 v_{I,j}^0 + \Delta P,$$

$$r_{i,j}^+ = \frac{v_{I-1,j} - v_{I,j}}{v_{I,j} - v_{I+1,j}},$$

$$r_{i,j}^- = \frac{v_{I+2,j} - v_{I+1,j}}{v_{I+1,j} - v_{I,j}},$$

$$r_{i-1,j}^+ = \frac{v_{I-2,j} - v_{I-1,j}}{v_{I-1,j} - v_{I,j}},$$

$$r_{i-1,j}^- = \frac{v_{I+1,j} - v_{I,j}}{v_{I,j} - v_{I-1,j}},$$

$$r_{I,J+1}^+ = \frac{v_{I,j-1} - v_{I,j}}{v_{I,j} - v_{I,j+1}},$$

$$r_{I,J+1}^- = \frac{v_{I,j+2} - v_{I,j+1}}{v_{I,j+1} - v_{I,j}},$$

$$r_{I,J}^+ = \frac{v_{I,j-2} - v_{I,j-1}}{v_{I,j-1} - v_{I,j}},$$

$$r_{I,J}^- = \frac{v_{I,j+1} - v_{I,j}}{v_{I,j} - v_{I,j-1}}.$$

$$\Psi(r) = \max \left[0, \min \left(2r, \frac{1+3r}{4}, \frac{3+r}{4}, 2 \right) \right],$$

$\alpha_x = 1$ if $F_x > 0$ and $\alpha_x = 0$ if $F_x < 0$, (no need to consider α_x if $F_x = 0$),

$$F_{I,J+1} = \frac{u_{i-1,J+1} + u_{i,J+1}}{2},$$

$$F_{I,J} = \frac{u_{i-1,J} + u_{i,J}}{2},$$

$$F_{i,j} = \frac{v_{I,j} + v_{I+1,j}}{2},$$

$$F_{i-1,j} = \frac{v_{I-1,j} + v_{I,j}}{2},$$

$$a_p^0 = \frac{\Delta L}{\Delta t},$$

$$\Delta P = p_{i-1,j} - p_{i,j}.$$

And by changing the flux limiter function Ψ above, the TVD scheme can adapt to any different alternative options, like if it's 0 it's just the form of upwind.

iii. Boundary Condition Treatment and Continuity Equation:

As for the treatment of boundary conditions, there're two types. The first one corresponds to the situation when the velocity component's cell edge is right at the boundary, here would adapt Leonard mirror node extrapolation to obtain possibly required nodal values outside the physical domain.

The second situation is that the velocity component's nodal position is right at the boundary, and in this case if the upwind direction is from the wall (then no further upwind point exists in physical domain), here would adapt the Hybrid Differencing scheme as an accuracy order degeneration treatment comparing with higher order scheme that however requires more points' information.

Depending on the Peclet Number, $P_e = \frac{|F_x|}{Re\Delta L}$, if it's smaller than 2, central difference

scheme is adapted in this edge, otherwise the upwind scheme is used. So this would modify the relevant term corresponding to this edge in above a_p , $a_{neighbour}$, and S_u coefficients. If it is the upwind, a_p and $a_{neighbour}$ don't need to change and just set corresponding face term in S_u as 0, then move that boundary neighbor also to the source term. If it is the central difference, still set corresponding face's term in the differenced-corrected source term as 0, and in this case the part $a_{p, \text{boundary}}$ in a_p and

$a_{neighbour, \text{boundary}}$ in $a_{neighbour}$ would then be:

$$a_{p,b} = \frac{1}{Re\Delta L} + \frac{F_+}{2}, \quad a_{neighbour,b} = \frac{1}{Re\Delta L} - \frac{F_+}{2}, \quad \text{if the boundary is in positive directions;}$$

$$a_{p,b} = \frac{1}{Re\Delta L} - \frac{F_-}{2}, \quad a_{neighbour,b} = \frac{1}{Re\Delta L} + \frac{F_-}{2}, \quad \text{if the boundary is in negative directions.}$$

And also move those known boundary value to the source term.

For the continuity equation used by SIMPLE to correct the pressure, as the zero-gradient boundary type is implemented for all four walls, here only needs to take all the interior pressure nodes as account and choose their cells as the volume evaluating the flow continuity. Its discretized form is then:

$$u_{i,J+1} - u_{i,J} + v_{I+1,j} - v_{I,j} = 0 \text{ (centered at } i,j).$$

Note that when introducing the pressure and corresponding velocity correction in SIMPLE loop, the boundary condition for p' is also zero-gradient. And the pressure of one point should be fixed at a reference value (like 0), otherwise the pressure equation would just evaluate the relevant values and insufficient to tell the exact solution.

II. Transient SIMPLE loop, Iterative and Elimination Solution Methods.

The idea of SIMPLE (Semi-Implicit Method for Pressure Linked Equations) is starting a loop from initially guessed velocity V_0 and pressure fields p^* , based on which decides the surface flux term F , pressure gradient term ΔP and other coefficients needed to calculate the momentum equations for a new velocity field V^* , just in the form of above discretized equations.

Also based on same coefficients, deriving the equation that relates the pressure correction p' and velocity correction V' . An approximation is made here as dropping the neighboring coefficients of V' equation to simplify p' - V' relation. In this project based on above equations, it is $u'_{i,J} = (p'_{i,j-1} - p'_{i,j})/a_{i,J}$ and $v'_{I,j} = (p'_{i-1,j} - p'_{i,j})/a_{I,j}$, together with aforementioned new velocity field one can have the corrected velocity participating into the continuity equation.

Thus the p' can be solved implicitly, following which set $p^* + p'$ as p^* and $V^* + V'$ as V_0 for next loop. Until the procedure converges, (in this project, this is reached if maximum difference of V between V_0 normalized by V_{lid} is less than a defined tolerance, saying, 1e-05), the results are obtained for a new time step.

It is worth mentioning that SIMPLE helps to couple velocity and pressure based on the divergence free property of velocity in incompressible flow. An alternative method also develops an additional velocity-pressure coupled equation based on and replacing the continuity equation to solve for V and p , namely the Poisson Pressure Equation. The comparison is not discussed here, just to see that both share the common idea as making the continuity hold at every step until results get converged.

To solve the momentum equation, as aforementioned, when adapting the TVD scheme, to make the matrix coefficients bounded, the higher order relations with downstream and further upstream neighbors (shown by the flux limiter function) need to be moved into the source term as the differenced correction. Therefore, the iterative method must be used solving the momentum equation, and here simply adapts the Jacobian Iteration, not as advanced like the multi-grid method but it works well.

To solve for the pressure correction equation, noting that the matrix system indeed satisfies the Scarborough criterion so as bounded. ($A_p = \sum A_{neighbour}$ for all pressure points except for the one fixed as the reference, equivalent as $A_p=1$ and $A_{neighbour}=0$.) And performing the iterative method as Gauss-Siedel, it indeed converges if the mesh is not refined. But as the mesh is refined, the relaxation parameter should be delicately adjusted and even so the convergence is reached slow and hard.

To trouble shoot, a possible reason would be if the boundary condition is specified here as zero-gradient, all the mesh points the central coefficient cannot weight over its neighboring but just equal, with the only one exception at a corner which is fixed as the reference. And by running a MatLab demo, even the location of such fixed reference would affect the converge speed when mesh is coarse. Thus moving this fixed value into the interior points would help a little bit but not obvious. When mesh is refined, this only one point's central coefficient overweight takes less fraction compared with the whole domain. And the simple iterative method can hard to converge.

	26	27	28	29	30	31
)	0	0	0	0	0	-0.1305
)	0	0	0	0	0	-0.0234
)	0	0	0	0	0	-0.0042
)	0	0	0	0	0	-7.5424e-04
)	0	0	0	0	0	-1.3537e-04
)	0	0	0	0	0	-2.4295e-05
)	0	0	0	0	0	-4.3602e-06
)	0	0	0	0	0	-7.8254e-07
)	0	0	0	0	0	-1.4045e-07
)	0	0	0	0	0	-2.5206e-08
)	0	0	0	0	0	-4.5238e-09
)	0	0	0	0	0	-8.1190e-10
)	0	0	0	0	0	-1.4571e-10
)	0	0	0	0	0	-2.6152e-11
)	0	0	0	0	0	-4.6936e-12
)	0	0	0	0	0	-8.4238e-13
)	0	0	0	0	0	-1.5119e-13
)	0	0	0	0	0	-2.7133e-14
)	0	0	0	0	0	-4.8726e-15
)	0	0	0	0	0	-8.7194e-16
)	0	0	0	0	0	-1.6242e-16
)	0	0	0	0	0	-2.5645e-17
)	0	0	0	0	0	-8.5484e-18
)	0	0	0	0	0	0
)	0	0	0	0	0	0
)	0	0	0	0	0	0
)	0	0	0	0	0	0
)	0	0	0	0	0	0
)	0	0	0	0	0	0
)	0	0	0	0	0	0.1088

Figure 3. Debug demo of non-convergence of p' equation by iterative method.

Interestingly, this non-convergence is found highly related with such boundary condition when the initial guess is not good. As debugging the output, when the maximum residual doesn't converge, it also doesn't diverge but always appears at the same location and with the same residual. This location is just the one influenced by corner of upper edge as the connection of the moving lid and static side walls. As shown in later result, this two points corresponds to singular points (the value increases as the mesh is refined), that end up hard to dealt with in this iterative method. It's also hard to think in real situation, would it be possible these corners are isolated from atmospherical pressure, and can the singular values really be maintained.

If it is the case that the pressure boundary is exposed to outside atmosphere as a fixed value, in the matrix there are lots more central coefficients weight over the neighboring (all four sides other than a single point), so the system is quite more bounded and the convergence is much easier to be reached from arbitrary initial guess.

Then eventually still considering the zero-gradient boundary condition, the pressure equation is here solved by the elimination method analogous to PDMA, as one central pressure point is related by the p' equation with n, s, w, e directions' neighboring. The idea is to forward represent former two unknowns by latter three starting from the upper left corner of the matrix where no further former unknowns, and continue this procedure to the bottom right corner where no further latter unknowns. And local equations start to be closed at the end, the solutions obtained are then substituted backward to close former ones, eventually the solution is got.

Differently, as the p' matrix is hard to be constructed exactly penta-diagonal, the assumption is made that in the interior domain the pressure won't vary that much for the ideal fluid case and static cavity. So the coefficients farther away from the diagonal are omitted (when sets the reference value as 0) initially forward, if they are not yet coupled with unknowns already met in forward elimination. The result would have slight difference, but in general is proved to be accurate as below comparing with result given by a commercial software.

For the SIMPLE loop, to pursue the result at different time, one just need to change the v^0 , u^0 and Δt in the source term and a_p . It takes time to let the SIMPLE loop converge at every chosen v^0 , u^0 from previous time step and a Δt . If mainly pursuing a steady state solution, it's equal to say let the time step Δt large enough to run the SIMPLE loop once. Otherwise, pursuing the detailed solution for every transient step would take much more time, especially when refining the mesh and Δt has to be even smaller to limit the courant number.

III. Results and Discussions.

i. Reynolds number analysis

To simulate the fluid case under different conditions, the Reynolds number is chosen to vary from 1, 10, 100 to 1000 with a grid size of 320^2 , and grid size (corresponds to number of P nodes) also varies as 40^2 , 80^2 , 160^2 , 320^2 for Reynolds number at 100. To provide a benchmark calculating the error on mesh density, a 500^2 grid size's simulation is run on OpenFOAM under same boundary and initial conditions, and which adapts linear upwind (SOU) scheme.

Plots are shown as following:

Grid Size 320^2 , Re 1:

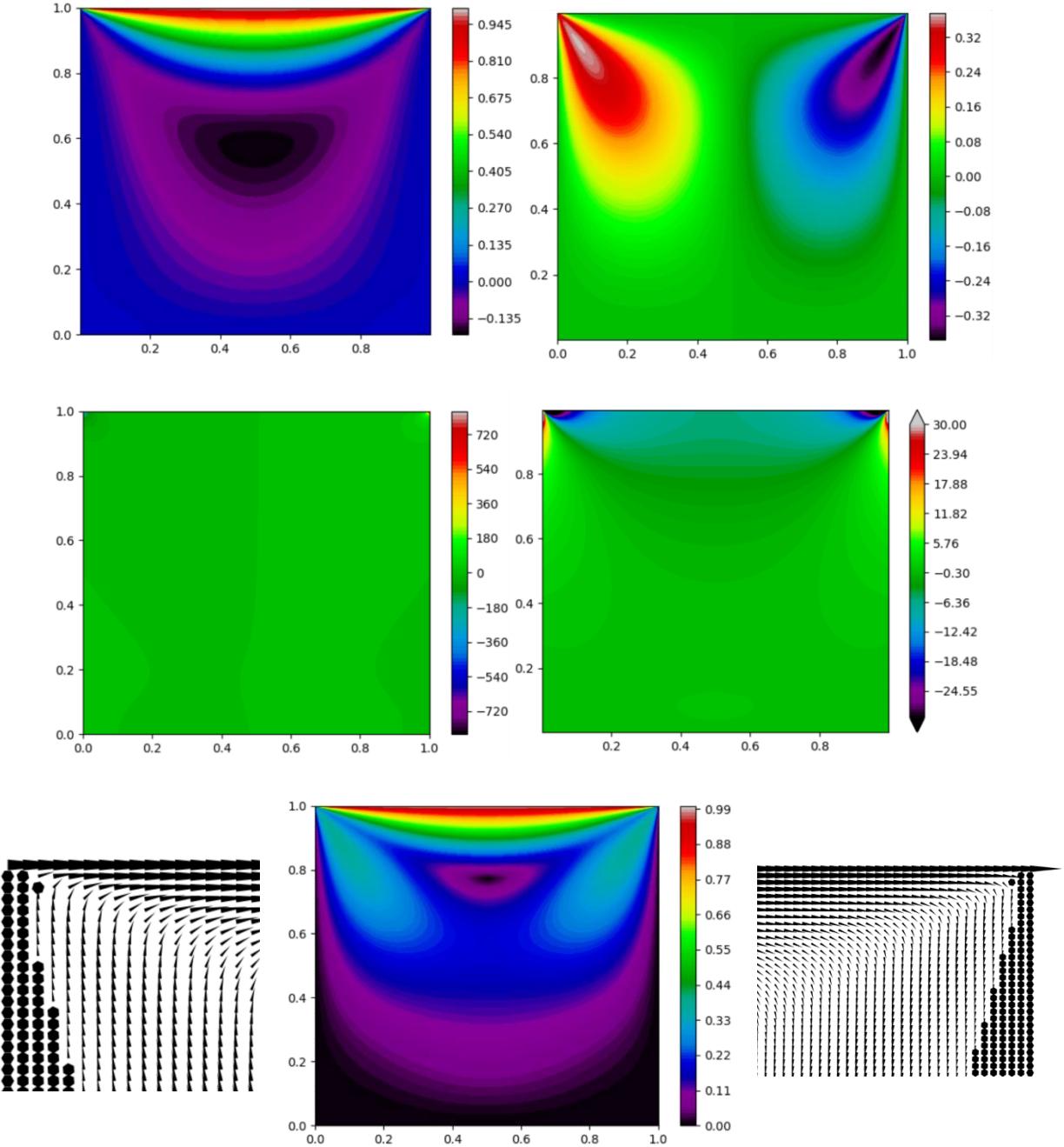


Figure 4. $u, v, p, \omega, |U|$ (upper left, upper right, middle left, middle right, bottom) contour profile in the cavity of grid size 320^2 and Re 1; vector plots are the left upper corner and right upper corner of vector U .

Fix the density as 1.0 kg/m^3 , cavity side length as 1.0 m , and dynamic viscosity as $1.0 \times 10^{-3} \text{ kg(ms)}^{-1}$, which is the hydrodynamic property quite closed to water under room temperature, Re 1 corresponds to a lid speed at 1 mm/s . This is actually adequately snow, and we could find that the profile is quite symmetric for both

directional components u , v (of course v has opposite signs at left and right edges) and velocity magnitude. The flow state is gentle and vortex is not very tense, and vorticity ω defined as $\frac{\partial v}{\partial y} - \frac{\partial u}{\partial x}$ is mostly slight expect at the two edges, that are obviously shown in the vector plots.

Note the black dots appearing are because the 320^2 is too big the size for computer (maybe the poor GPU) or the plot tool (python Matplotlib) to deal with the quiver plotting, so it would use those black dots to replace too small arrows standing for small velocity magnitude. Seeing Fig. 5, black dots area just corresponds to low magnitude in Fig. 4 bottom picture, where if sticking the ratio of arrow length as velocity magnitude, lower part of the cavity would have too small arrows to display.

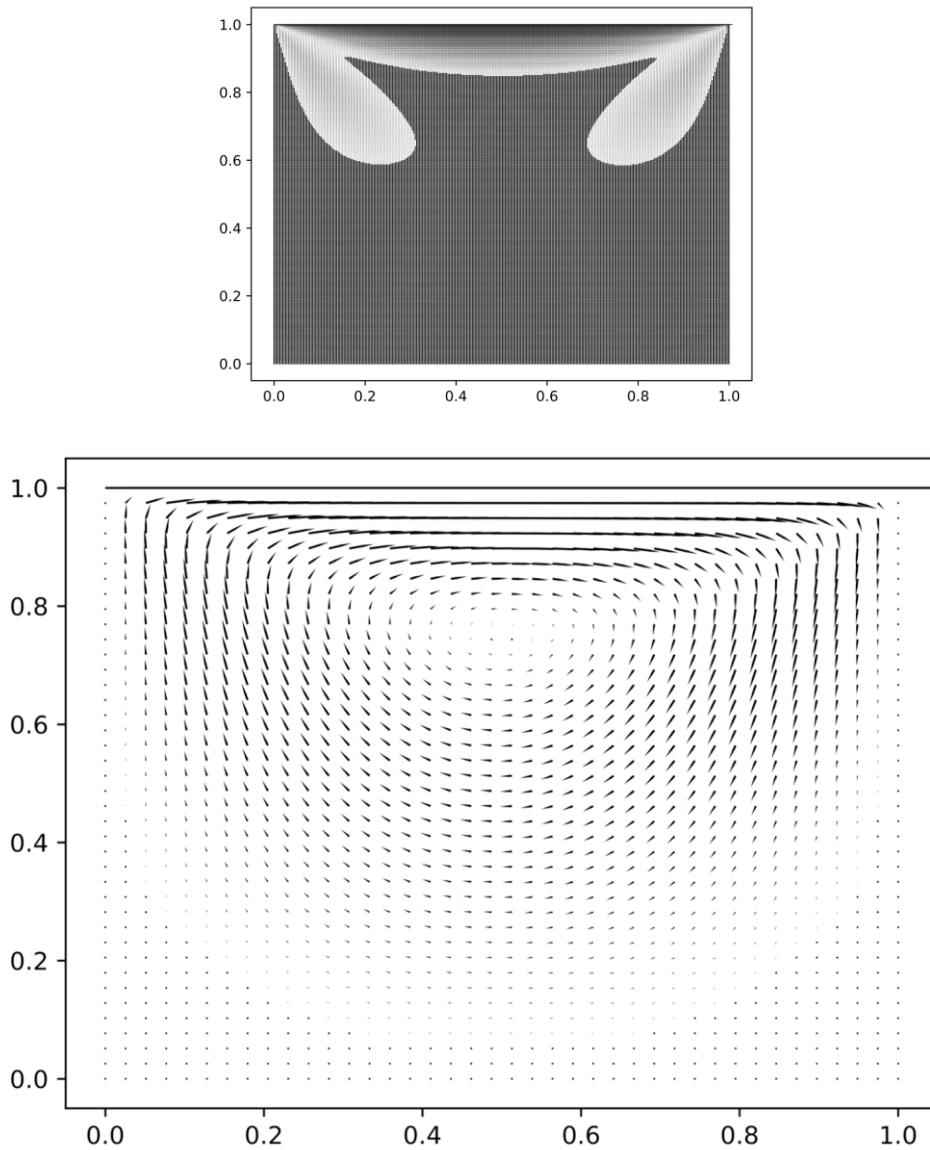


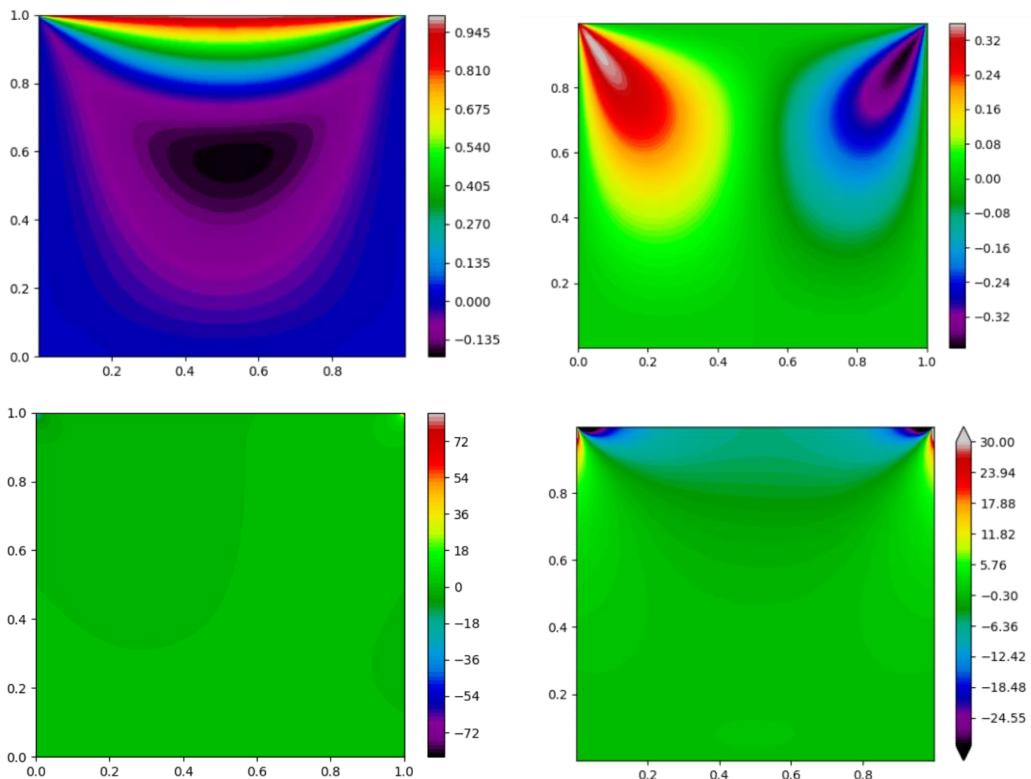
Figure 5. The grid points are too dense to display the vector well, small magnitude region are just replaced by black dots, corresponding to Fig.4 velocity magnitude profile. The one from size 40^2 is also shown for better visualization.

Clear pictures could be found in the attached file (about 12 M), or in below section when the mesh size is reduced to 80 or 40, the vector plot would be more clear, showing the vortex phenomenon. So in Fig. 4 bottom the scaled view of Fig. 5's two corners, the flow is restricted by the side wall and moving lid, so as the vorticity is the largest here but not the magnitude as the vortex would appearing in the center.

And as stated, pressure doesn't vary much in the middle, but just the upper tow corners. From below comparison, its trend is increasing as Re number decreases and mesh size increase, especially for increased mesh size, the pressure doesn't converge. So it might as well just remember the pressure tends to be positively large at the right end of the lid and negatively large at the left end of the lid, but the variation of the extreme value exists in the infinitesimal small scale, as every time the mesh is refined this two corners' value increase. It is then not necessary to take these two corner's pressure into more consideration, so the error analysis below would mainly focus on u , v difference and leave out these two singular points of pressure.

To see how the Re number would affect the steady results, more plots are shown below.

Grid Size 320^2 , Re 10:



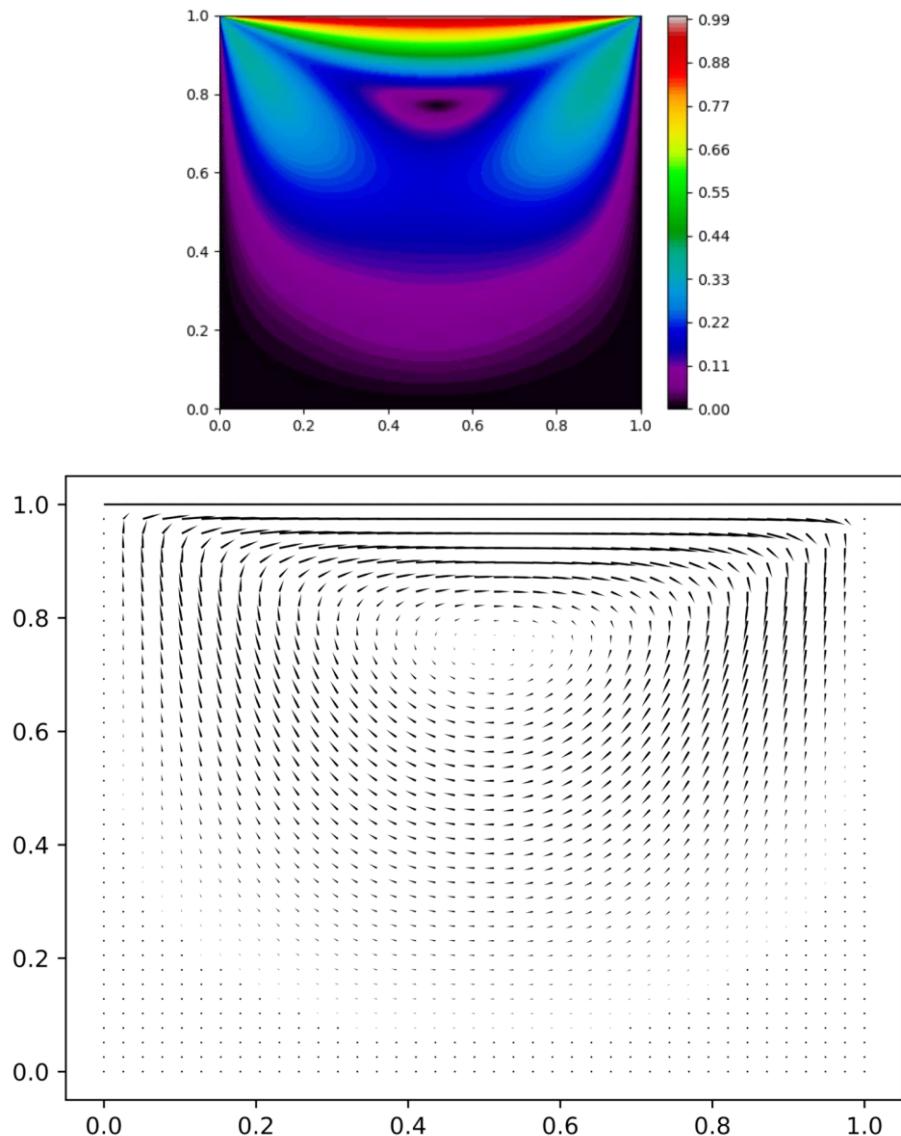


Figure 6. $u, v, p, \omega, |U|$ (upper left, upper right, middle left, middle right, bottom) contour profile in the cavity of grid size 320^2 and $Re = 10$; vector plot is from the size of 40^2 .

Similar, this would stand 1cm/s lid velocity for water inside the cavity. Generally, the variation of above quantities is not much from $Re = 1$, except that the edge points' pressure magnitude decreases a level, but inside is still about 0. From all 3 velocity profiles, a very little trend is found that symmetry starts not being preserved, and the maximum downward velocity increase slightly. Still the vortex is not strong and the stationary center (in Fig. 4 and Fig. 6 bottom picture at upper center the black region of 0 velocity) doesn't develop towards the right side.

Grid Size 320^2 , $Re = 100$:

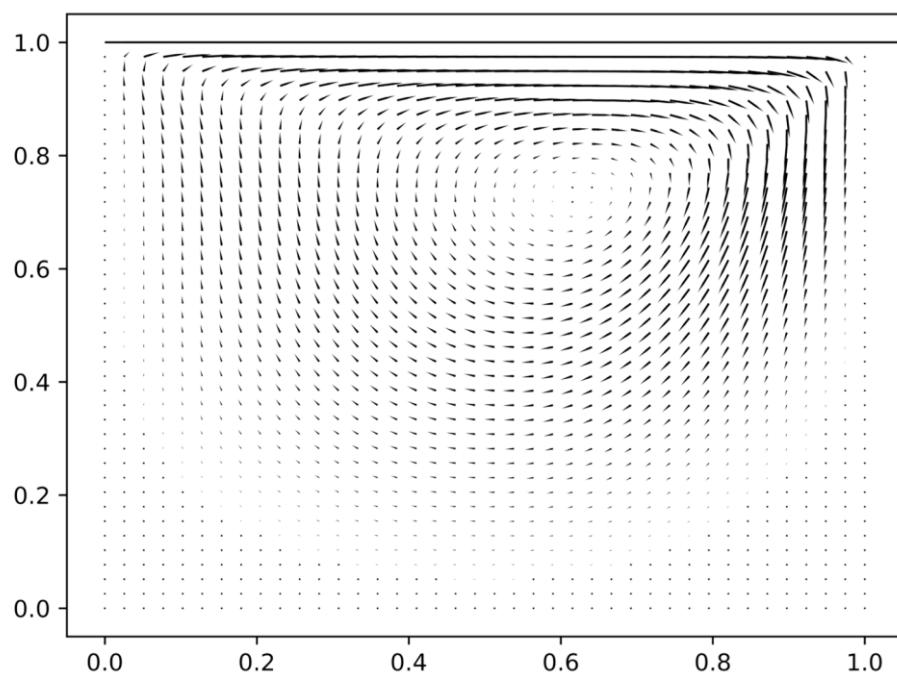
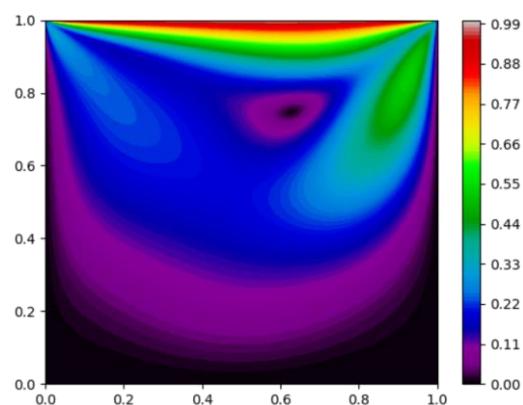
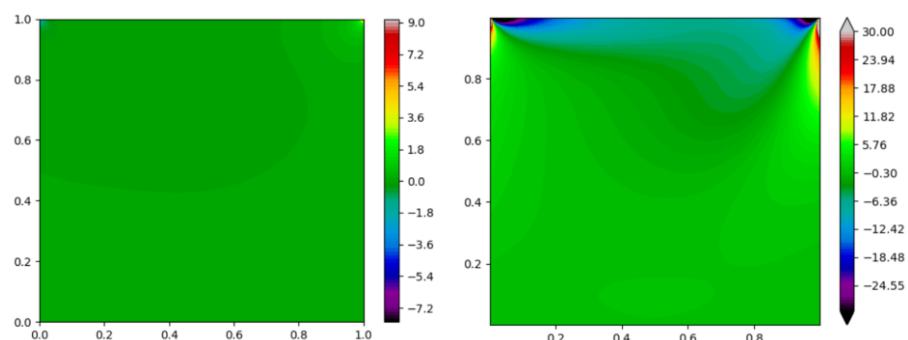
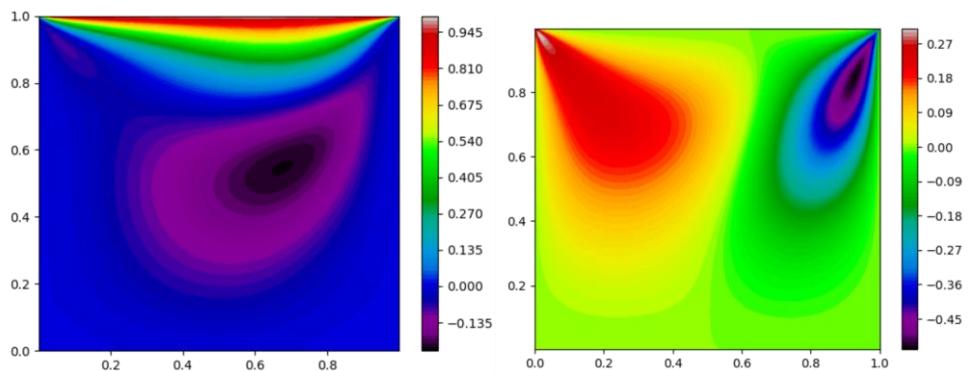
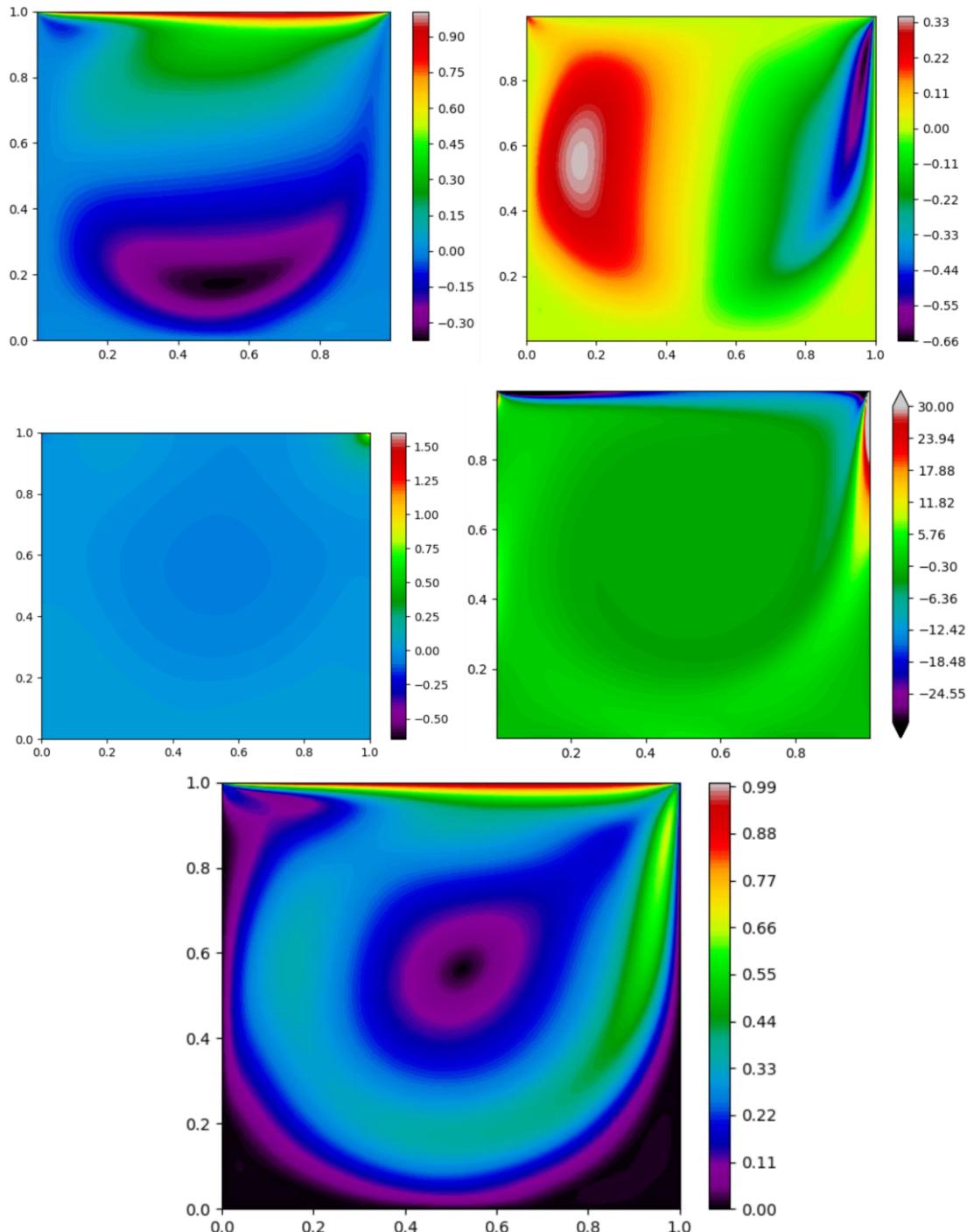


Figure 7. u , v , p , ω , $|U|$ (upper left, upper right, middle left, middle right, bottom) contour profile in the cavity of grid size 320^2 and $Re = 100$; vector plot is from the size of 40^2 .

When the Reynolds number increases to 100, as the lid velocity of 1 dm/s, the vortex phenomenon becomes more obvious and asymmetric. Vorticity distribution also changes, the tilting for velocity towards left wall also appears at the left upper corner. The water in the cavity is now more influenced by the moving lid and the space for bottom water to remain stationary is also compressed from two sides. The vector plot above from grid size 40^2 can also show this clearly.

Grid Size 320^2 , $Re = 1000$:



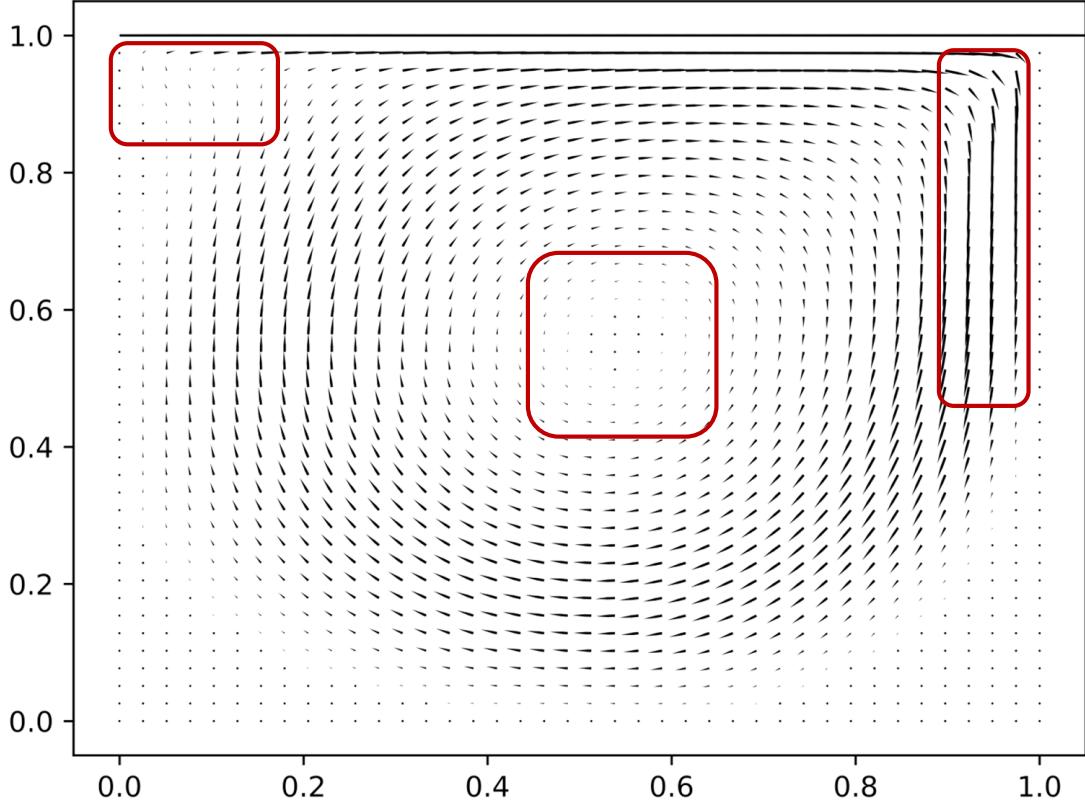


Figure 8. u , v , p , ω , $|U|$ (upper left, upper right, middle left, middle right, bottom) contour profile in the cavity of grid size 320^2 and $Re = 100$; vector plot is from the size of 40^2 .

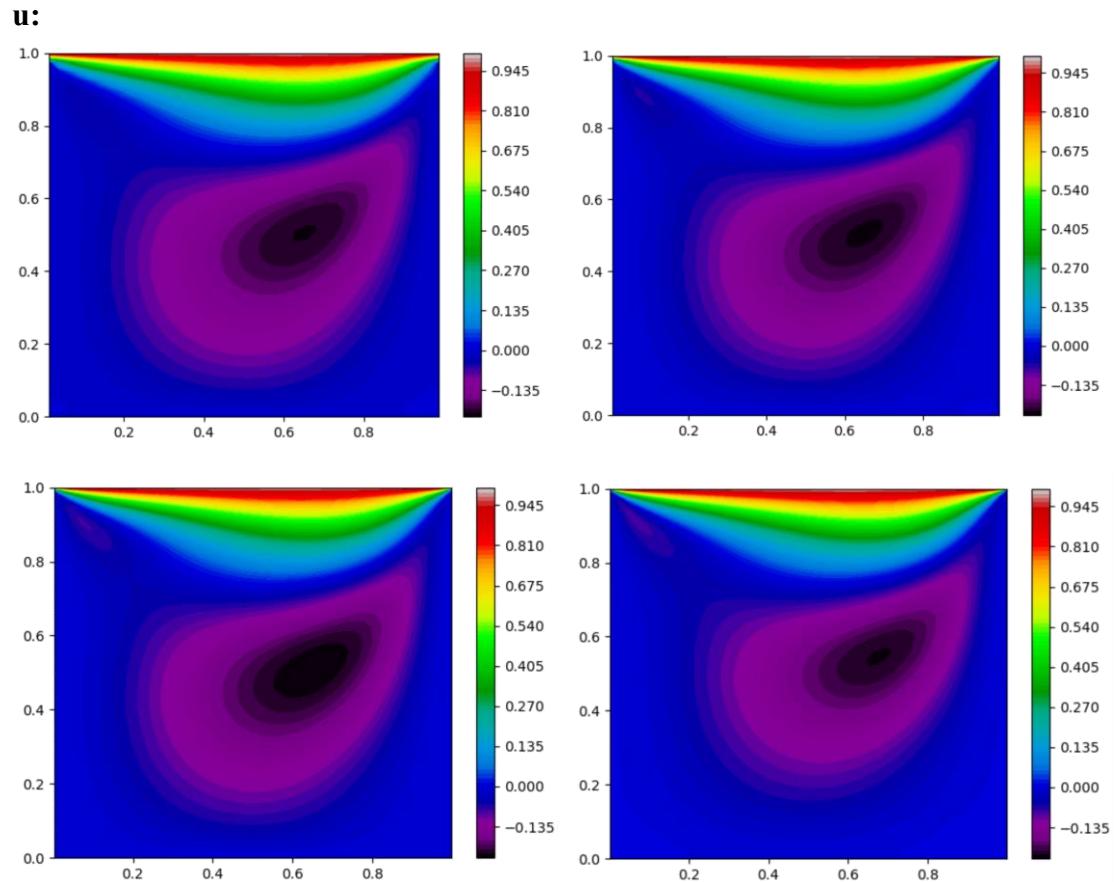
When Re increases to 1000 as the water situation of 1m/s lid speed, the situation becomes quite different. First, the bottom region of the cavity is much influenced, and less area of stationary could remain. Second, as the lid velocity is now quite fast, the upper left region has more negative u directions as the viscosity is now small compare with the convection that it can't attach closely downward layers to move rightwards, so the sense of separation occurs at the left upper corner and by static pressure gradient it drags fluid downside towards the corner so here're left-upper pointing arrows. Third, the middle vortex center position is more downside compared with low Re cases, and due to more intense fluid carried by the lid bump against the right wall, so both the negative v and vorticity here are much larger than previous cases, even this continues to more downside area. The vorticity distribution changes especially a lot as the vortex move downwards, circles are found in above contour plot. Though the pressure is still around 0 in the middle of the cavity but the variation in this case is more obvious, especially for the part of the upper right corner, the increase pattern is now more obvious.

Besides above discussion, one of the most obvious difference under different Reynolds number shown by the code running and simulation is its time consumption to reach the steady state. Generally speaking, larger Re number would take more time for the flow in the cavity to reach steady, and here takes the example of grid size 80^2 to demonstrate this trend. For the real time needed from the simulation results, it takes about 0.172152s when Re equals 1, 1.82278s when Re equals 10, 13.9392s when Re equals 100, 34.2835s when Re equals 1000, all of which are from the same initial condition as pure static flow.

ii. Grid Size analysis

To find the effect as the mesh is refined, here fixes the Reynold's number as 100, and varies the mesh size from 40, 80, 160 to 320, also the CPU time is recorded. And to set a benchmark, here uses OpenFOAM icoFoam solver to compute a 500^2 size's linear upwind with same initial and boundary conditions. To calculate the error, here first interpolates the result fields form OF to the same size of each above, and compute the error field as their difference. Then the maximum error is definitely expected at the upper two corners, so for more information, the error field is also plotted and a cell-averaged value is also calculated based on the number density on the absolute difference.

Following are the results:



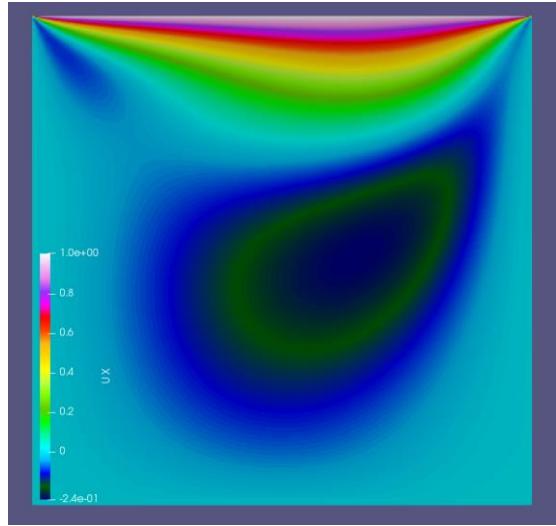


Figure 9. Grid size of 40^2 , 80^2 , 160^2 , 320^2 's (upper left, upper right, middle left, middle right) contour plots of \mathbf{u} and an OF reference result of 500^2 .

As mesh is refined, the minimum value of u is predicted better and the upper left corner's tilting is more obvious, and stationary vortex center is more concentrated. For extreme values, it is clearer in below plots of the error field defined as above fields from this code minus the last reference one.

u_error:

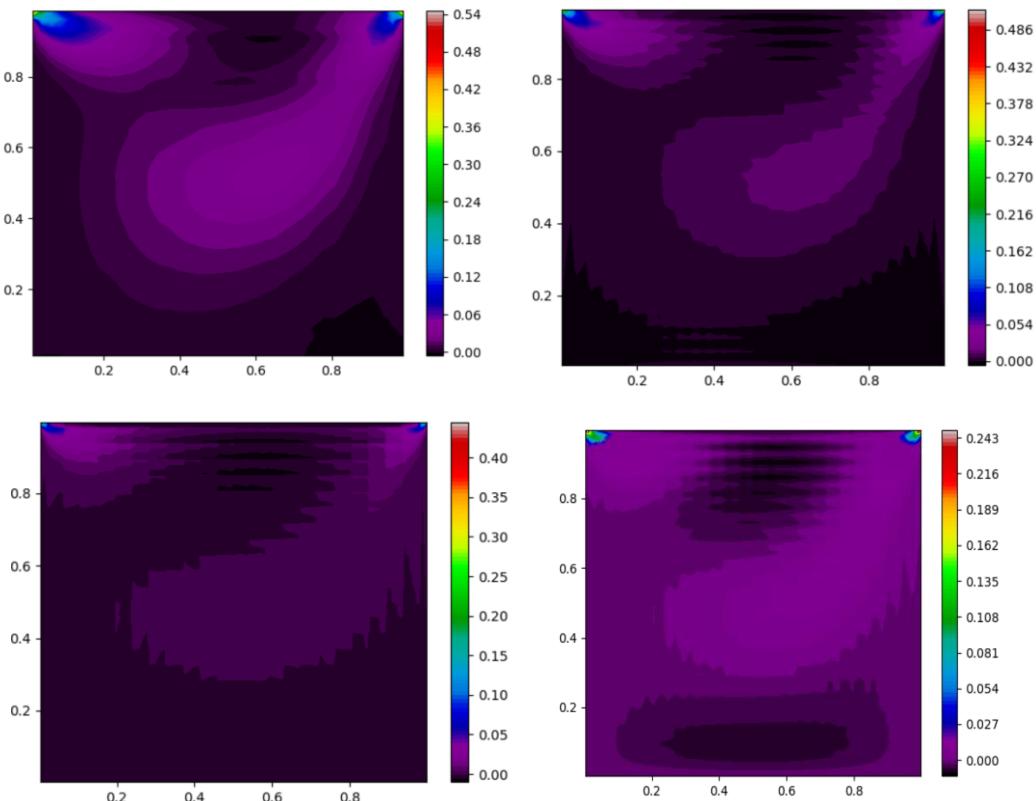


Figure 10. The error field plots of u at grid size of 40^2 , 80^2 , 160^2 , 320^2 (upper left, upper right, lower left, lower right).

To have an overview sense of the error change, a cell averaged one is calculated for above errors, and this result gives:

Grid size 40^2 : 0.01372890570408284

Grid size 80^2 : 0.005827838214893447

Grid size 160^2 : 0.002518621224223484

Grid size 320^2 : 0.0012535156673994992

v:

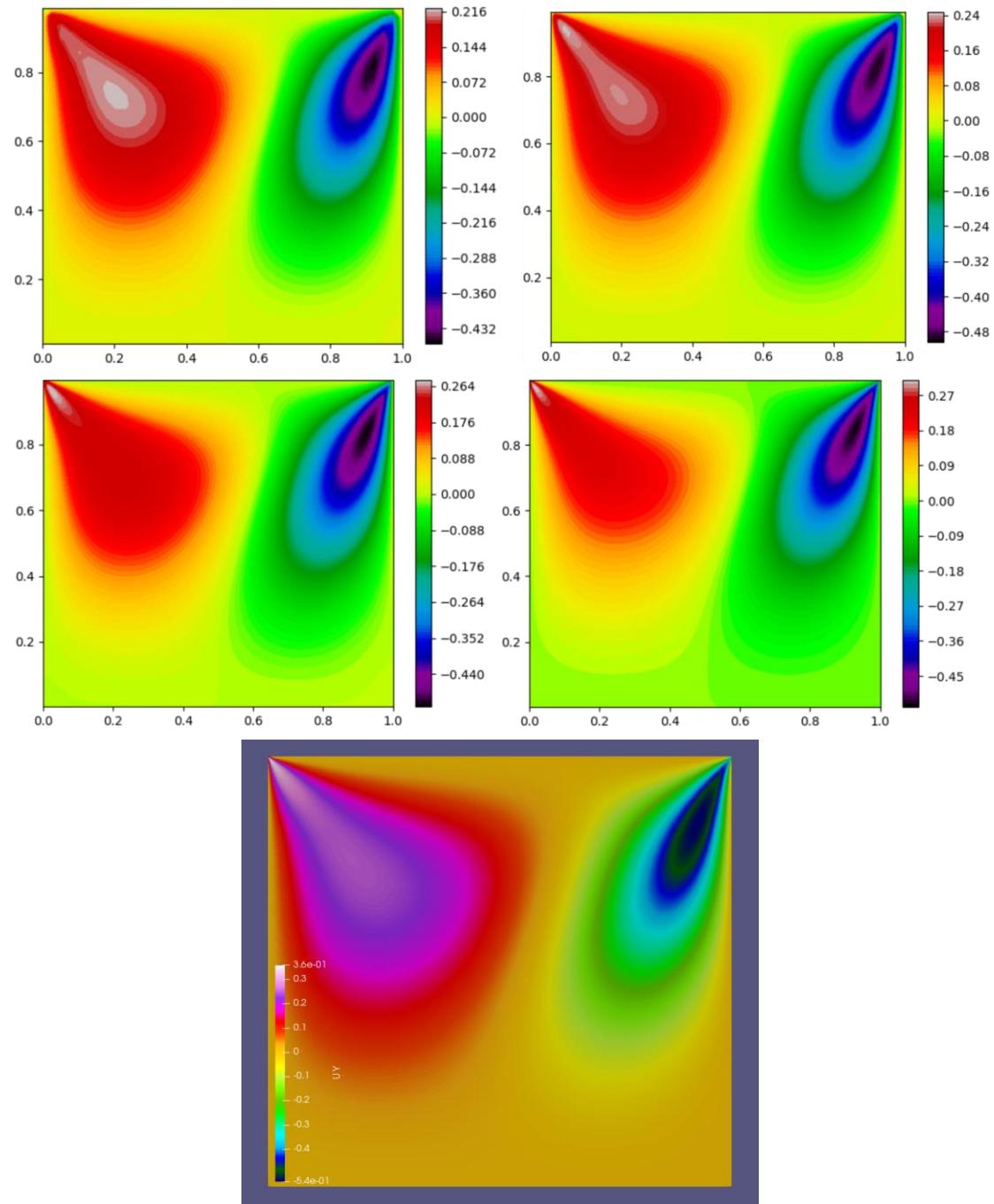


Figure 11. Grid size of 40^2 , 80^2 , 160^2 , 320^2 's (upper left, upper right, middle left, middle right) contour plots of \mathbf{v} and an OF reference result of 500^2 .

The most noticeable change is that as the mesh is refined, the location where the maximum positive v value occurs moves from a middle left area (about 0.2, 0.7) to the left corner of the lid, which is correctly indicated by OF results. Namely, the size of 40^2 and 80^2 cannot address this well until further refined to denser mesh. Similarly, the error field is drawn below.

v_error:

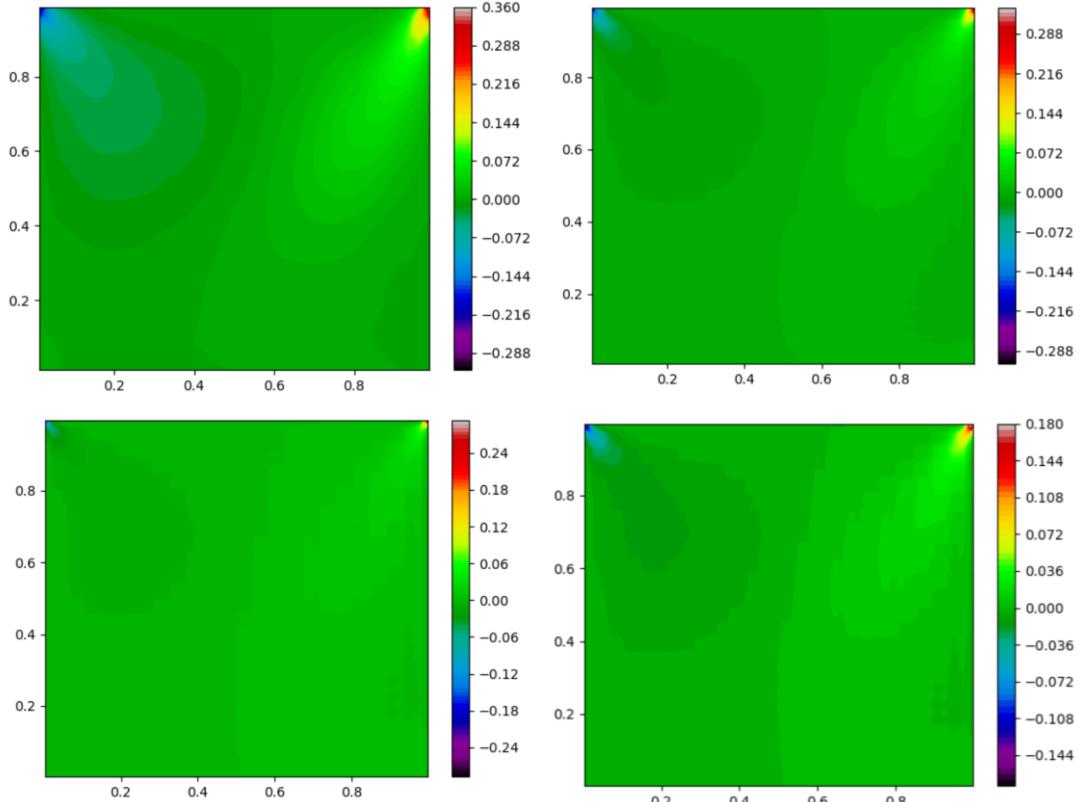


Figure 12. The error field plots of v at grid size of 40^2 , 80^2 , 160^2 , 320^2 (upper left, upper right, lower left, lower right).

Still, a cell averaged one is calculated for above errors, and this result gives:

Grid size 40^2 : 0.013601629091226825

Grid size 80^2 : 0.006700611718036004

Grid size 160^2 : 0.003910908055272782

Grid size 320^2 : 0.0009607322621874798

P:

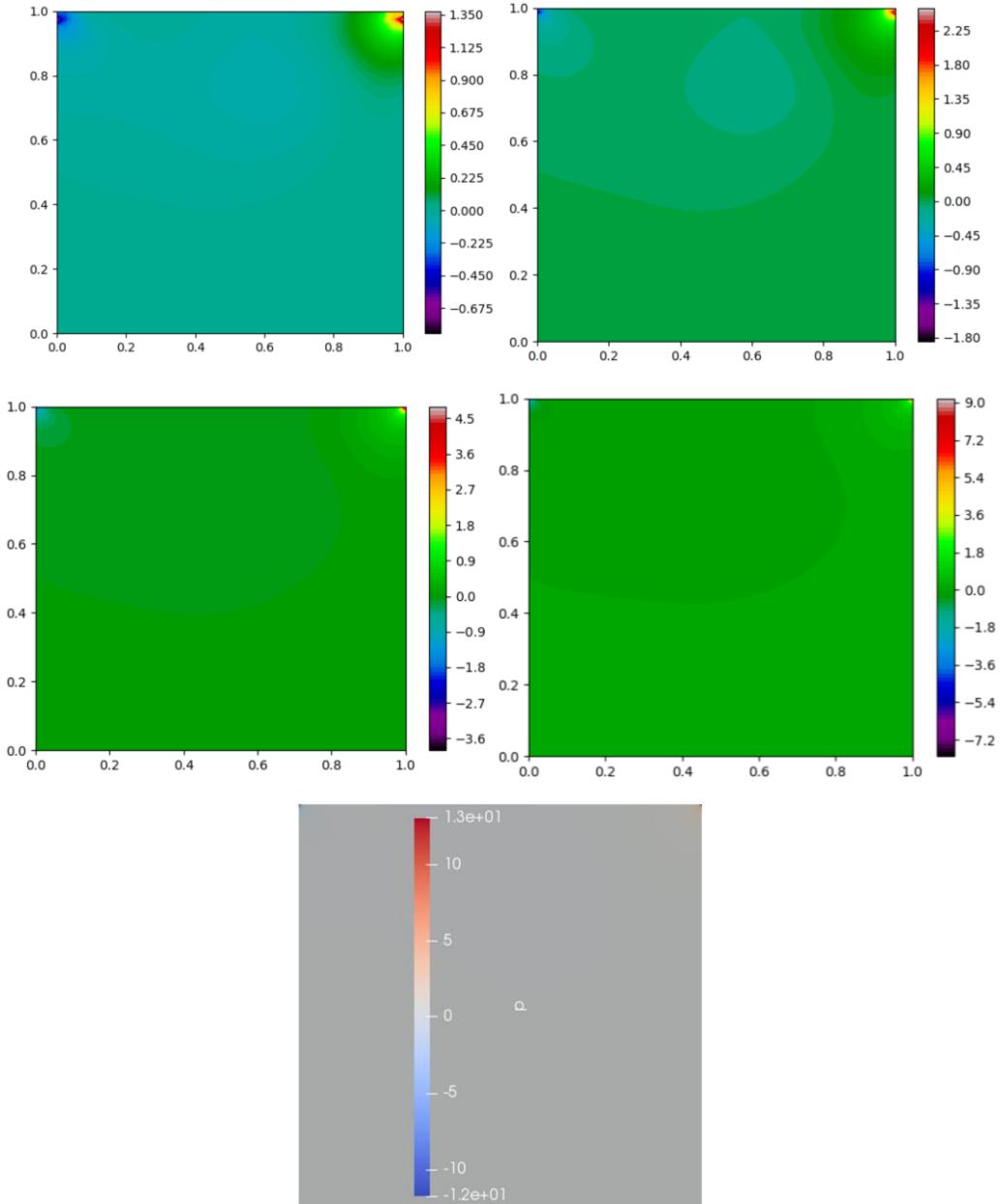


Figure 13. Grid size of 40^2 , 80^2 , 160^2 , 320^2 's (upper left, upper right, middle left, middle right) contour plots of \mathbf{p} and an OF reference result of 500^2 .

As stated before, the pressure of upper two corners keep increasing as the mesh is refined and is the case in both this code and OF demo. But a common trend is that except those two corners, interior pressure field tends to be universally zero as mesh is further refined in this Re number. To show that mesh density can affect the corner pressure very much, a case demo is run by OF with Re equaling 10 of grid size 50^2 and 100^2 shown in Fig. 14.

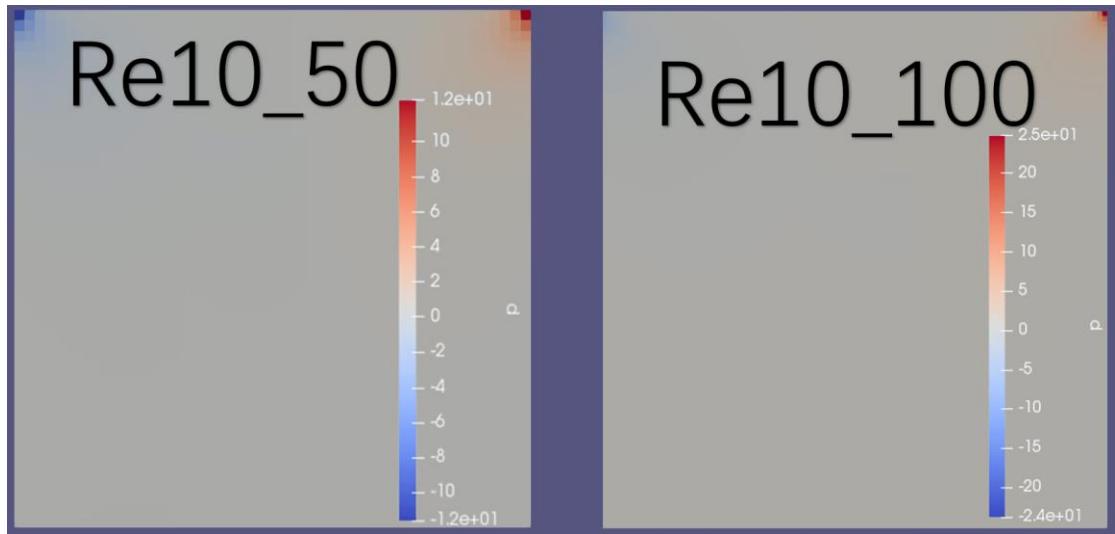


Figure 14. The effect of grid size on extreme values of pressure field.

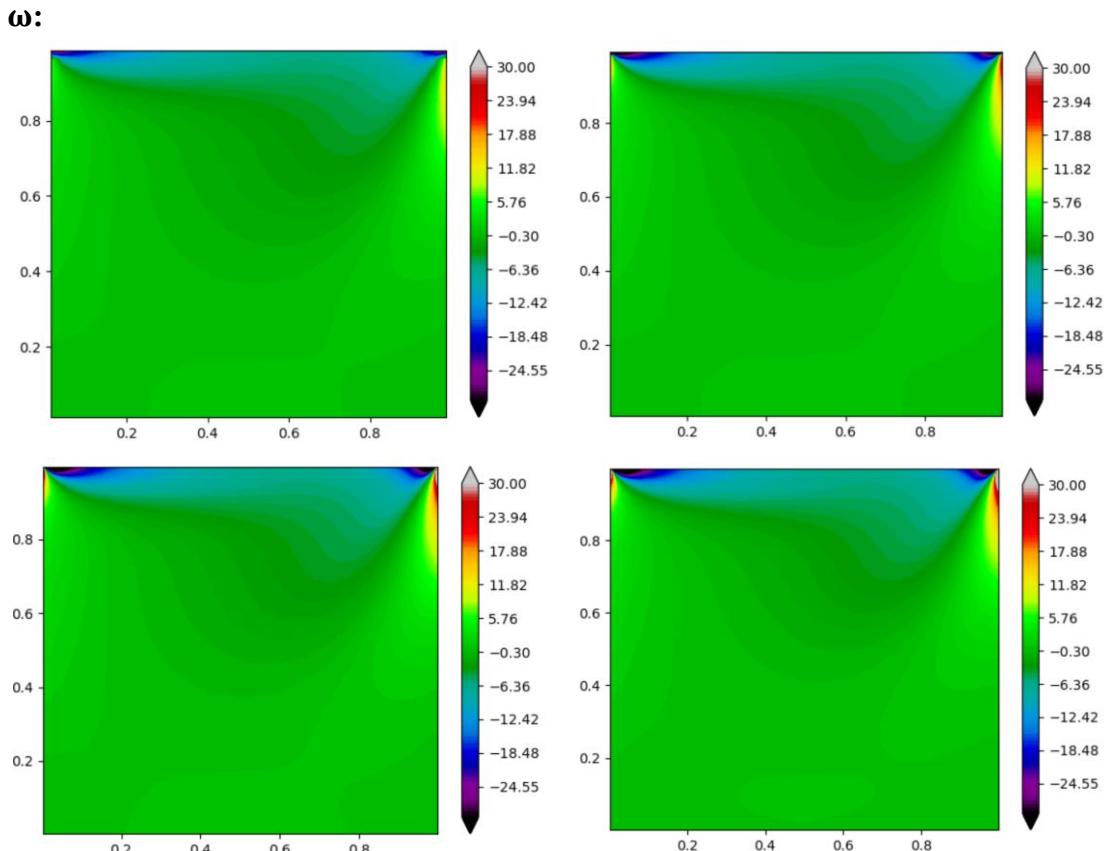


Figure 15. Grid size of 40^2 , 80^2 , 160^2 , 320^2 's (upper left, upper right, middle left, middle right) contour plots of ω .

In vorticity profile, the mainly change in refined mesh is that the vorticity near two upper corners,

which corresponds the forced turning of flow directions by the moving lid and side wall, is resolved better. But in the middle, the vorticity distribution doesn't change a lot.

Vector Plot of \mathbf{U} :

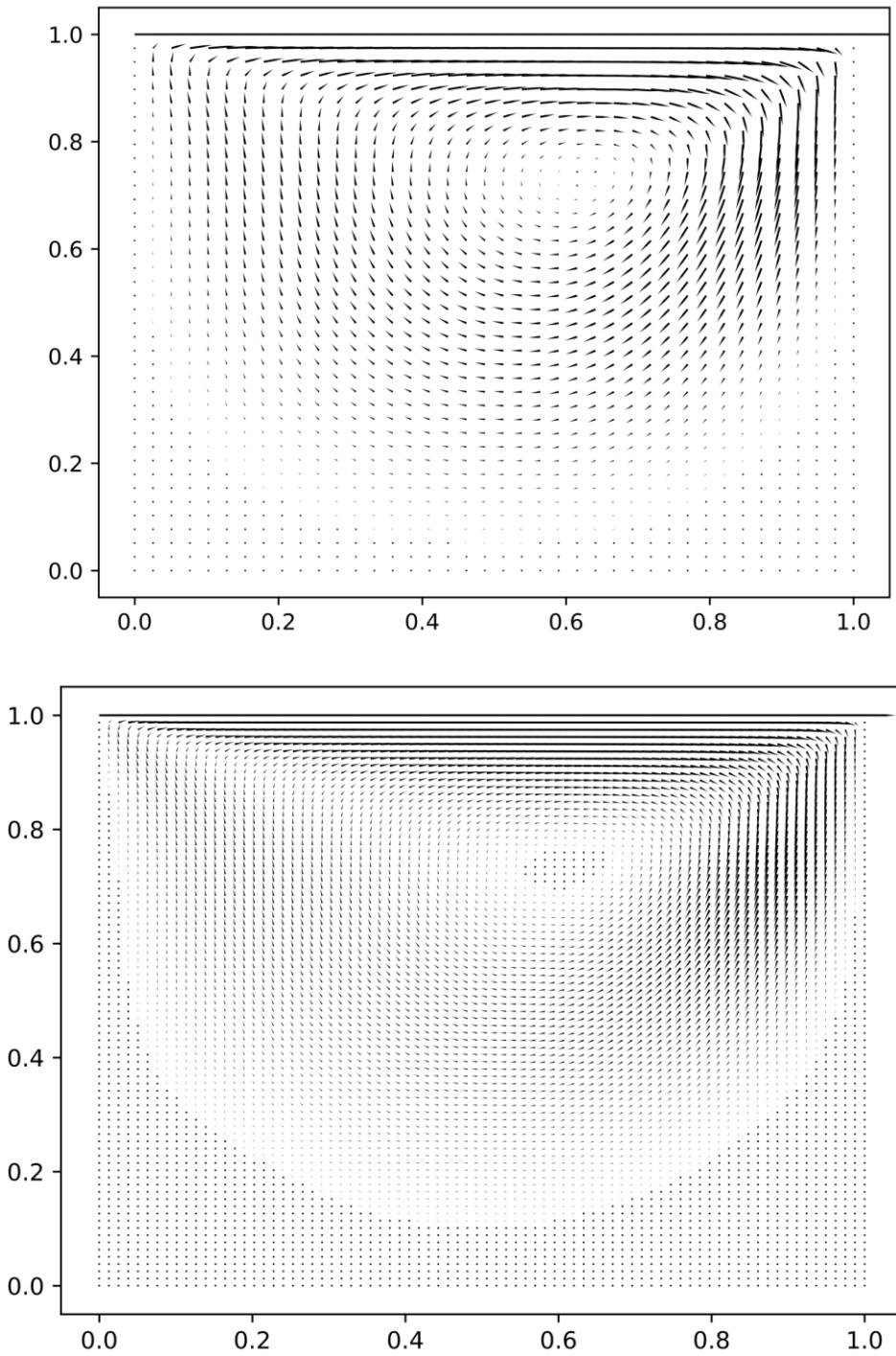


Figure 16. Velocity vector plots of grid size 40^2 and 80^2 , higher size is too dense to show, and dots stand for values as velocity magnitude that are small.

CPU time comparison:

Test conditions: Re 100, relaxation for p in SIMPLE loop 0.1, relaxation for u, v in SIMPLE loop 0.3, relaxation for iterative solution method 0.9.

Grid size 40^2 : 641.107 ms

Grid size 80^2 : 6678.7 ms

Grid size 160^2 : 64795.4 ms

Grid size 320^2 : 599299 ms

It can be found that the CPU time cost increases not just as the order of N^2 , which turns out to be 4 times each, but it is in fact around 10 times. This is mainly due to that as the mesh is refined, the convergence damping is obviously found.

Here changing the relaxation for iterative solution method doesn't help a lot as each SIMPLE loop the time cost is exactly about 4 times. But for the outer loop of SIMPLE, if the relaxation for u, v, p doesn't decrease the whole solution would diverge.

For 40 and 80 sizes, increasing the relaxation could further speed up the code, but for 160 and 320, where the residual damps up and down approaching the final tolerance one, the relaxation needs further decrease to speed up a little bit.

And just increasing the tolerance a little bit, the whole loops needed might be cut to a half. The damping is one kind of this simple iterative method, and could possibly helped by advanced gradient method where the correction is more directional.