AM5630 - Foundations of CFD

Assignment -3

Lid Driven Cavity using Stream Function vorticity approach

Vijay S (AM16D025)
Yogeshwaran R (AM16D028)
Department of Applied Mechanics
IIT Madras, Chennai

Table of Contents

List of Figures

1. Problem Statement

- Using stream function vorticity approach, solve the lid driven cavity problem where the upper plate is moving with velocity of 1.0 m/s in positive x direction as shown below.

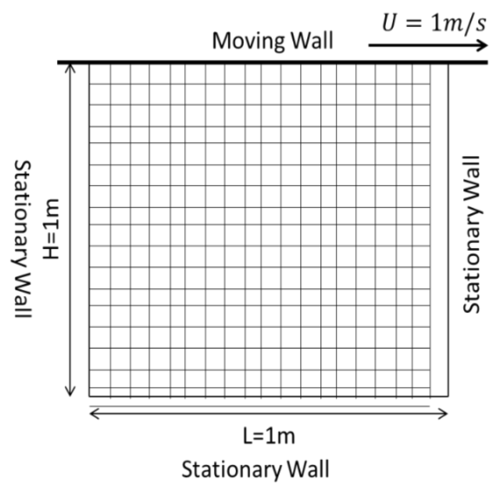


Figure 1 Lid Driven closed cavity

- The cavity is of size 1m x 1m. Take the convergence criteria for the stream function equation as CONGS = 0.001; and to achieve a steady state solution, CONSS is set to 0.002.
- Compare/Validate the numerical results with Ghia et al., "*High-Re Solutions for Incompressible Flow Using the Navier-Stokes Equations and a Multigrid Method", Journal Of Computational Physics*, Vol. 48, pp387-411.
- Plot streamlines, vorticity, velocity vector & centerline velocity profile for the Reynold's number (Re) = 400 and 1000.
- Extend the above problem to lid driven cavity with open flow channel as shown below.

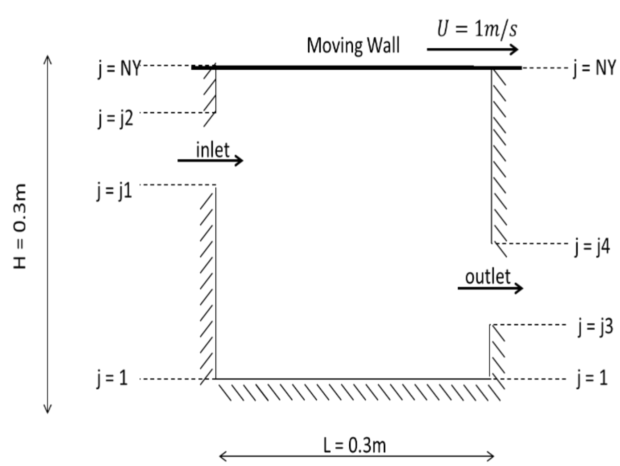


Figure 2 Lid Driven cavity with inlet & outlet ports

2. Governing Equation:

(Continuity Equation) $\frac{\partial^2 \varphi}{\partial x^2} + \frac{\partial^2 \varphi}{\partial y^2} = -\omega$ (1)

(Vorticity transport Equation) $\frac{\partial \omega}{\partial t} + u\frac{\partial \omega}{\partial x} + v\frac{\partial \omega}{\partial y} = \frac{1}{Re}(\frac{\partial^2 \omega}{\partial x^2} + \frac{\partial^2 \omega}{\partial y^2})$ (2)

(Velocity Components) $u = \frac{\partial \varphi}{\partial y}, \quad v = -\frac{\partial \varphi}{\partial x}$ (3)

(Pressure poison Equation) $\frac{\partial^2 p}{\partial x^2} + \frac{\partial^2 p}{\partial y^2} = 2\rho\left[\left(\frac{\partial^2 \omega}{\partial x^2}\right)\left(\frac{\partial^2 \omega}{\partial y^2}\right) - \left(\frac{\partial^2 \omega}{\partial x\,\partial y}\right)^2\right]$ (4)

3. Discretization of Governing Equation:

$$-\omega_{i,j} = \frac{\varphi_{i+1,j} - 2\varphi_{i,j} + \varphi_{i-1,j}}{\Delta x^2} + \frac{\varphi_{i,j+1} - 2\varphi_{i,j} + \varphi_{i,j-1}}{\Delta y^2} + O(\Delta x^2, \Delta y^2) \quad (5)$$

$$\omega_{i,j}^{n+1} = \omega_{i,j}^{n} + \Delta t * \left[\begin{matrix} \frac{1}{Re}\left(\frac{\omega_{i+1,j} - 2\omega_{i,j} + \omega_{i-1,j}}{\Delta x^2} + \frac{\omega_{i,j+1} - 2\omega_{i,j} + \omega_{i,j-1}}{\Delta y^2}\right) \\ -\left(u_{i,j}\frac{\omega_{i+1,j} - \omega_{i-1,j}}{2\Delta x} + v_{i,j}\frac{\omega_{i,j+1} - \omega_{i,j-1}}{2\Delta y}\right) \end{matrix}\right] \quad O(\Delta x^2, \Delta y^2, \Delta t) \quad (6)$$

$$u = \frac{\varphi_{i,j+1} - \varphi_{i,j-1}}{2\Delta y} + O(\Delta y^2), \quad v = \frac{\varphi_{i+1,j} - \varphi_{i-1,j+1}}{2\Delta x} + O(\Delta x^2) \quad (7)$$

$$p_{i,j} = 0.25\left[p_{i+1,j} + p_{i-1,j} + p_{i,j+1} + p_{i,j-1}\right] - 0.5\rho\Delta x^2\Delta y^2\left[\begin{matrix}\left(\frac{\omega_{i+1,j} - 2\omega_{i,j} + \omega_{i-1,j}}{\Delta x^2}\right)\left(\frac{\omega_{i,j+1} - 2\omega_{i,j} + \omega_{i,j-1}}{\Delta y^2}\right) \\ -\left(\frac{\omega_{i+1,j+1} + \omega_{i-1,j-1} - \omega_{i+1,j-1} + \omega_{i-1,j+1}}{4\Delta x\Delta y}\right)^2\end{matrix}\right] \quad (8)$$

4. Boundary condition

4.1 Lid Driven Closed Cavity Problem

4.1.1 Vorticity Boundary Condition

4.1.1.1 Left wall

Consider Equation (1) along the surface, the stream function is constant, and its value may be specified arbitrarily; for example, $\varphi_{i,j} = \varphi_1$. Then, along left wall,

$\left(\frac{\partial^2 \varphi}{\partial y^2}\right)_{1,j} = 0$, and Equation (1) is then reduced to $\left(\frac{\partial^2 \varphi}{\partial x^2}\right)_{1,j} = -\omega_{1,j}$ (9)

Using Taylor's series, the second order derivative in the equation above is obtained as

$$\left(\frac{\partial^2 \varphi}{\partial x^2}\right)_{1,j} = \frac{2(\varphi_{2,j} - \varphi_{1,j})}{\Delta x^2} + O(\Delta x) \quad (10)$$

Substitution of Equation (10) into Equation (9) yields

$$\omega_{1,j} = \frac{2(\varphi_{1,j} - \varphi_{2,j})}{\Delta x^2} + O(\Delta x) \quad (11)$$

4.1.1.2 Right wall

A similar procedure is used to derive the boundary conditions at right boundary; the appropriate expression is given by,

$$\omega_{IM,j} = -\left(\frac{\partial^2 \varphi}{\partial x^2}\right)_{IM,j} = \frac{2(\varphi_{IM,j} - \varphi_{IMM1,j})}{\Delta x^2} + O(\Delta x) \quad (12)$$

4.1.1.3 Bottom wall

Since the bottom wall is fixed, the above procedure is followed for the bottom boundary also. The appropriate expression is given by,

$$\omega_{i,1} = -\left(\frac{\partial^2 \varphi}{\partial y^2}\right)_{i,1} = \frac{2(\varphi_{i,1} - \varphi_{i,2})}{\Delta y^2} + O(\Delta y) \quad (13)$$

4.1.1.4 Top wall

The upper surface is moving to the right with a constant velocity $u_0 = \left(\frac{\partial \varphi}{\partial y}\right)_{i,JM} = 1$m/s. Following the procedure described previously, the Taylor's series expansion at point (1, JMM1) yields,

$$\varphi_{i,JMM1} = \varphi_{i,JM} - \left(\frac{\partial \varphi}{\partial y}\right)_{i,JM} \Delta y + \left(\frac{\partial^2 \varphi}{\partial y^2}\right)_{i,JM} \frac{\Delta y^2}{2} + O(\Delta y^3),$$ from which

$$\omega_{i,JM} = \frac{2(\varphi_{i,JM} - \varphi_{i,JMM1})}{\Delta y^2} - \frac{2u_0}{\Delta y} + O(\Delta y) \quad (14)$$

4.1.2 Stream Function Boundary Condition

4.1.2.1 Left wall

Consider Equation (3), using Taylor's series, the first order x-derivative in the equation is obtained as $\left(\frac{\partial \varphi}{\partial x}\right)_{1,j} = \frac{4\varphi_{2,j} - \varphi_{3,j} - 3\varphi_{1,j}}{2\Delta x} + O(\Delta x^2)$, Since the wall is fixed, $\left(\frac{\partial \varphi}{\partial y}\right)_{1,JM} = 0$ and $\varphi_{i,j}$ at wall can be arbitrarily specified as mentioned earlier. Therefore, the above equation can be rewritten as

$$\varphi_{2,j} = \frac{1}{4}\left(\varphi_{3,j} - \varphi_1\right) + O(\Delta x^2), \quad (15)$$

4.1.2.2 Right wall

Similarly, using Taylor's series backward differencing for the right fixed wall, the equation obtained as,

$$\varphi_{IMM1,j} = \frac{1}{4}\left(\varphi_{IMM2,j} - \varphi_1\right) + O(\Delta x^2) \quad (16)$$

4.1.2.3 Bottom wall

Consider Equation (3), using Taylor's series, the first order y-derivative in the equation is obtained as $\left(\frac{\partial \varphi}{\partial y}\right)_{i,1} = \frac{\varphi_{i,2} - \varphi_{i,1}}{\Delta y} + O(\Delta y)$, Since $\left(\frac{\partial \varphi}{\partial y}\right)_{i,1} = 0$ and $\varphi_{i,1}$ can be any arbitrary value, the above equation becomes $\varphi_{i,2} = \varphi_1$ (17)

The arbitrary constant φ is set to zero for all four walls as there were no flow rate across the boundaries

4.1.2.4 Top wall

Similarly, for top wall, the Taylor's series for the first order y-derivative is obtained as $\left(\frac{\partial \varphi}{\partial y}\right)_{i,JM} = \frac{\varphi_{i,JMM1} - \varphi_{i,JM}}{\Delta y} + O(\Delta y)$, Since the upper boundary is moving, $\left(\frac{\partial \varphi}{\partial y}\right)_{i,JM} = u_0$, the above equation becomes

$$\varphi_{i,JM} = u_0 \Delta y + \varphi_{i,JM-1} + O(\Delta y), \qquad (18)$$

4.2 Lid Driven Cavity with inlet & outlet ports

The procedure followed in this case is as same as the closed lid driven cavity problem at all fixed regions. The boundary conditions for vorticity and stream function at inlet and outlet port are described in the following sections.

4.2.1 Vorticity Boundary condition

4.2.1.1 Inlet port

Utilizing approximation (Equation (10)) and a second-order central difference expression for the y-derivative and substituting in Equation (1),

$$\omega_{i,J} = \frac{2(\varphi_{1,j} - \varphi_{2,j})}{\Delta x^2} - \frac{\varphi_{1,j+1} - 2\varphi_{1,j} + \varphi_{1,j-1}}{\Delta y^2} + O(\Delta x, \Delta y^2), \qquad J1 < J < J2 \qquad (19)$$

4.2.1.2 Outlet port

Similarly, at the outlet, a simple extrapolation may be used for which one sets $\frac{\partial \varphi}{\partial x} = 0$ to provide

$$\omega_{i,J} = \frac{1}{3}\left(4\varphi_{IMM1,j} - \varphi_{IMM2,j}\right) + O(\Delta x^2) \qquad J3 < J < J4 \qquad (20)$$

4.2.2 Stream Function Boundary Condition

4.2.2.1 Inlet Port

The stream function at inlet is determined from the interior as provided below,

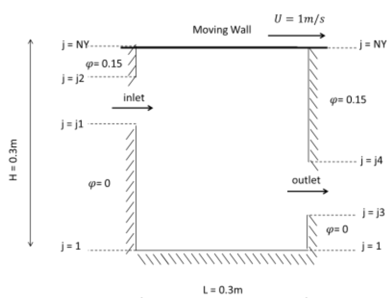
$$\varphi_{1,j} = \frac{1}{3}\left(4\varphi_{2,j} - \varphi_{3,j}\right) + O(\Delta x^2) \qquad J1 < J < J2 \qquad (21)$$

4.2.2.2 Outlet Port

Similarly, the stream function at outlet is determined as given below,

$$\varphi_{IM,j} = \frac{1}{3}\left(4\varphi_{IMM1,j} - \varphi_{IMM2,j}\right) + O(\Delta x^2) \qquad J3 < J < J4 \qquad (22)$$

4.2.2.3 Stationary Walls



5. Algorithm:

5.1 Pseudo Code

Step 1. Initialize $\boldsymbol{\varphi}, \boldsymbol{\omega}, \boldsymbol{u}, \boldsymbol{v}, \boldsymbol{p}$

Step 2. Compute the boundary conditions for vorticity & Stream function.

Step 3. Obtain $\boldsymbol{\varphi}$ by solving Poisson equation (Eqn. (5)) for the initial guess $\boldsymbol{\omega}_{i,j}^{k} = 0$, using Point successive over relaxation scheme with Convergence criteria of CONSF = 0.001

Step 4. Calculate $u_{i,j}$ and $v_{i,j}$ from equation (7)

Step 5. Calculate $\boldsymbol{\omega}_{i,j}^{k+1}$ from Vorticity equation (Eqn. (6))

Step 6. Update $\boldsymbol{\omega}_{i,j}^{k} = \boldsymbol{\omega}_{i,j}^{k+1}$

Step 7. Repeat the above steps $\boldsymbol{\omega}$ reached convergence (CONSS = 0.002)

Step 8. Calculate Pressure using equation (8)

5.2 Flowchart

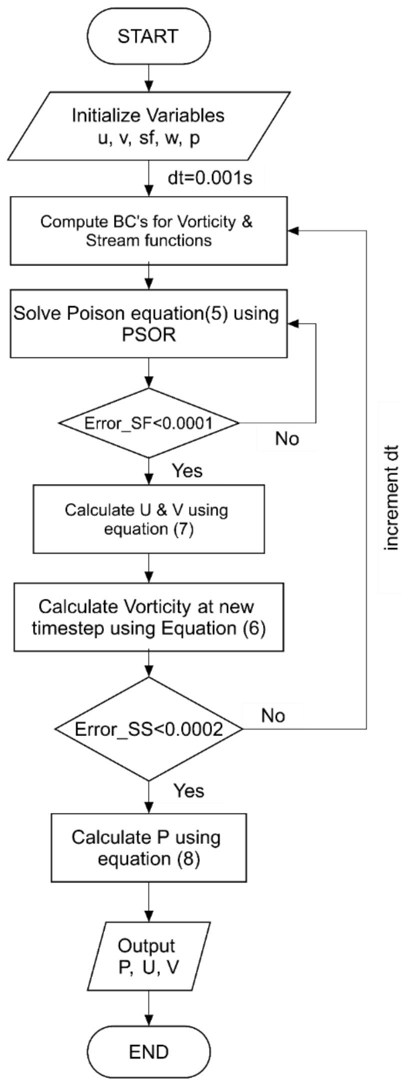


Figure 3 Flowchart of Stream Function Vorticity approach

6. Results and Discussion

6.1 Lid Driven Cavity:

6.1.1 Case 1- (Re = 1000)

Figure 4 and Figure 9 represents the variation of U-component velocity profile w.r.t 'y' along the centerline for Re=1000 and Re=400 respectively. The numerical results obtained using stream function vorticity approach has been compared/validated with results from Ghia et al and CFD analysis using FLUENT. The results are in excellent agreement with each other as shown in the figure below.

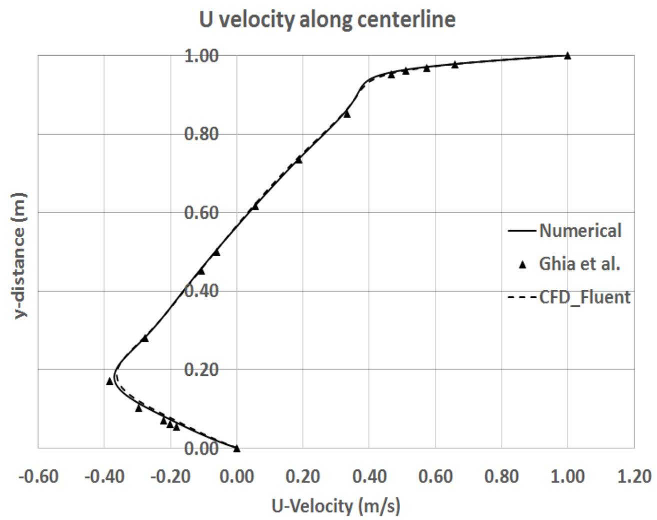


Figure 4 Comparison of U Velocity profile along x-centerline for RE=1000 (Numerical Code vs Ghia et al. vs CFD analysis)

Similar comparison for V-component velocity profile w.r.t 'x' along the centerline is compared in Figure 5 and 10 for Re=1000 and Re=400 respectively. The numerical code results are in excellent agreement with CFD solution and Ghia et al results.

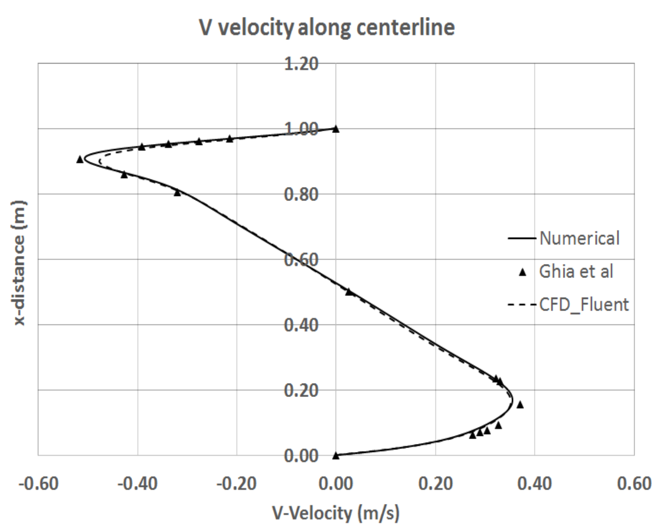


Figure 5 Comparison of V Velocity profile along y-centerline for RE=1000 (Numerical Code vs Ghia et al. vs CFD analysis)

The stream lines pattern, u and v component velocity contour, vorticity contours for Re=1000 and Re=400 are compared with Ghia et al and CFD in the figure 6,7,8,11,12,13 respectively. The stream patterns and velocity contours are in good qualitative and quantitate agreement with each other.

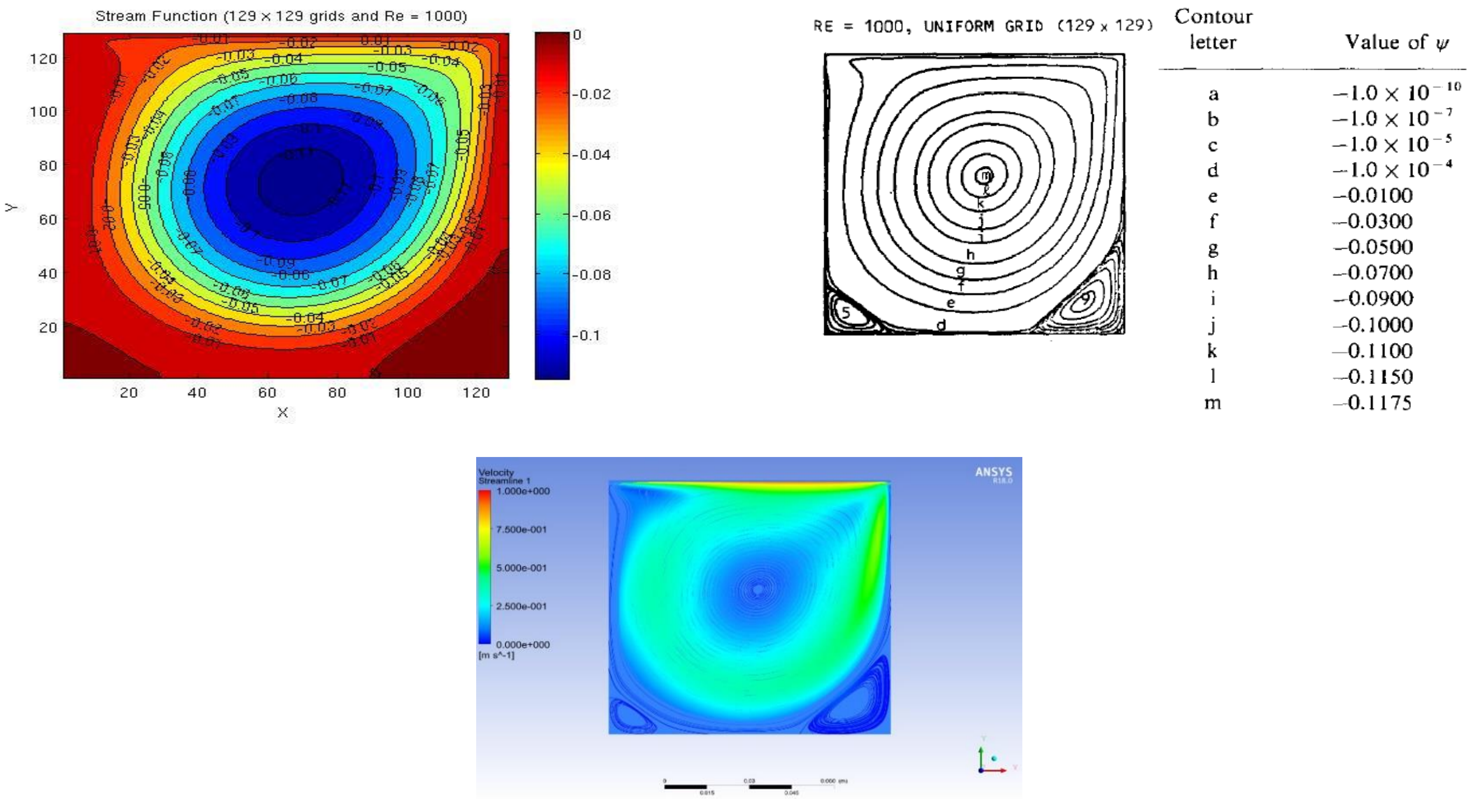


Figure 6 Comparison of Streamlines pattern for RE=1000 (Numerical Code vs Ghia et al. vs CFD analysis)

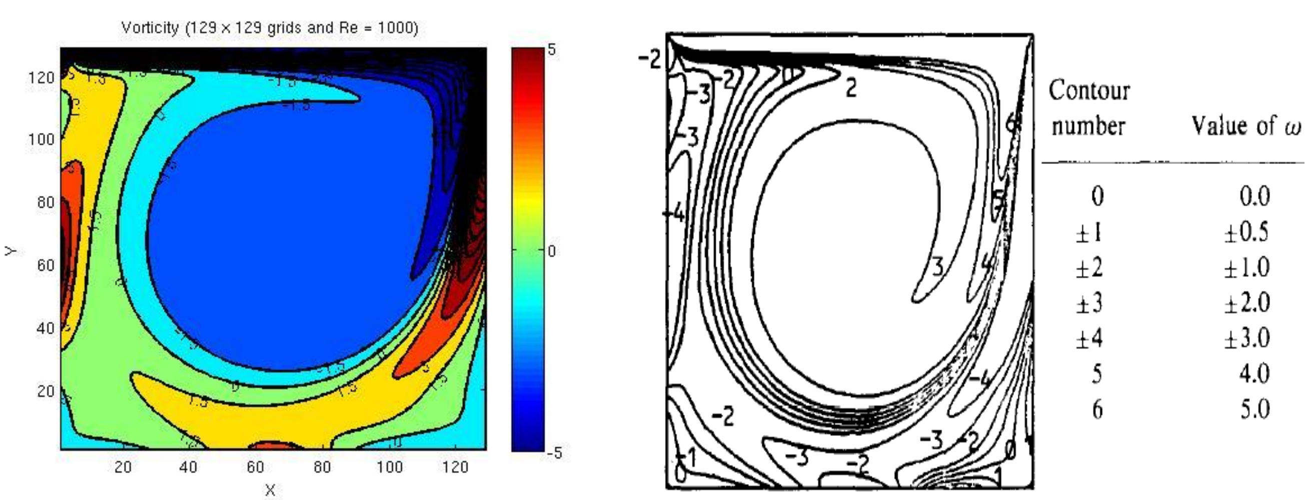


Figure 7 Comparison of Vorticity contours for RE=1000 (Numerical Code vs Ghia et al)

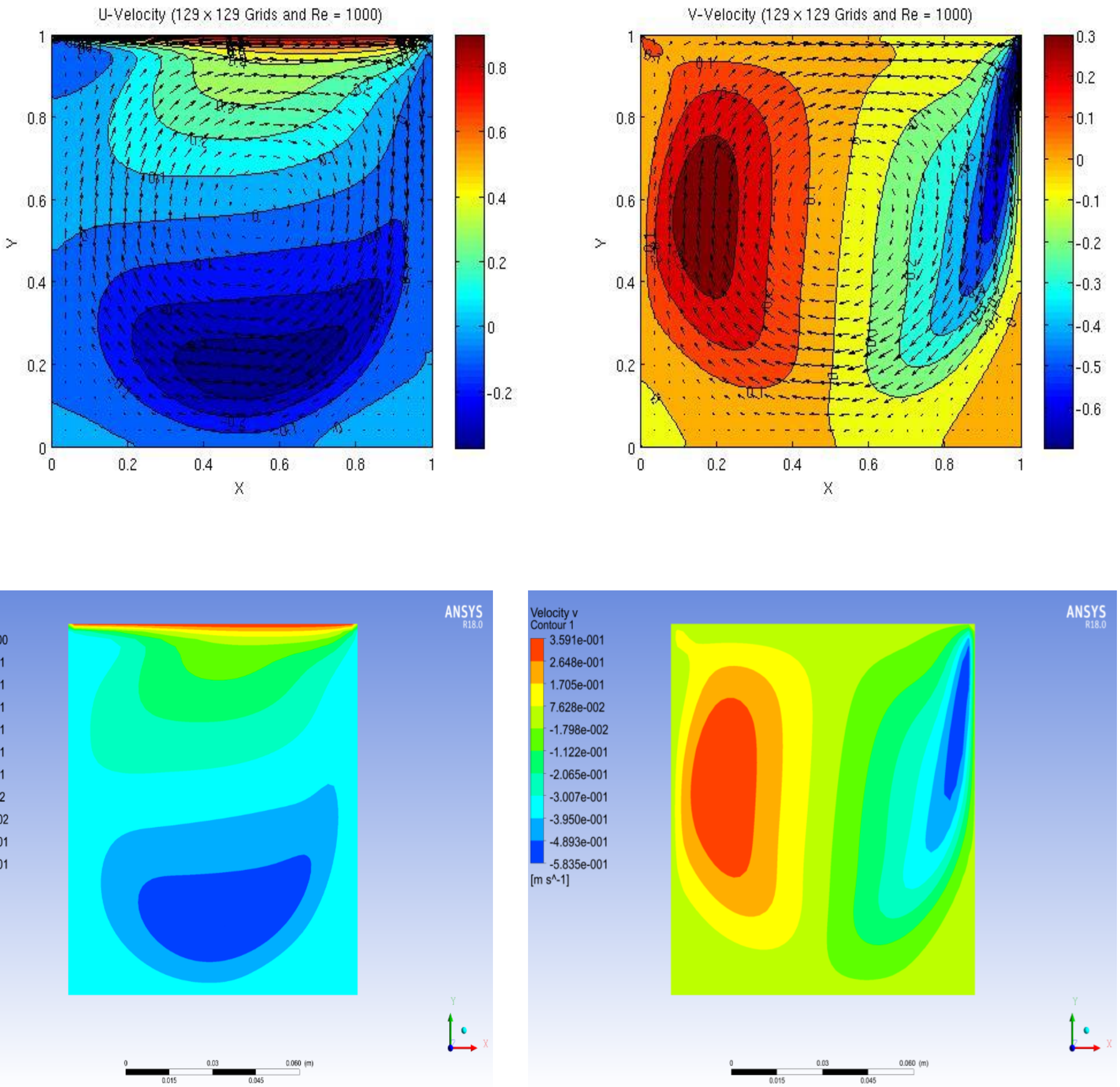


Figure 8 Comparison of U & V velocity contours for RE=1000 (Numerical Code vs CFD analysis)

6.1.2 Case 2- (Re = 400)

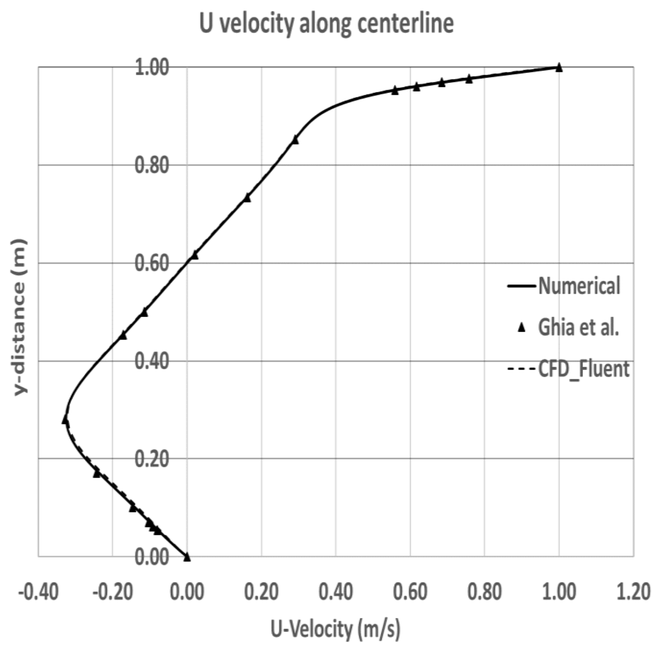


Figure 9 Comparison of U Velocity profile along x-centerline for RE =400 (Numerical Code vs Ghia et al. vs CFD analysis)

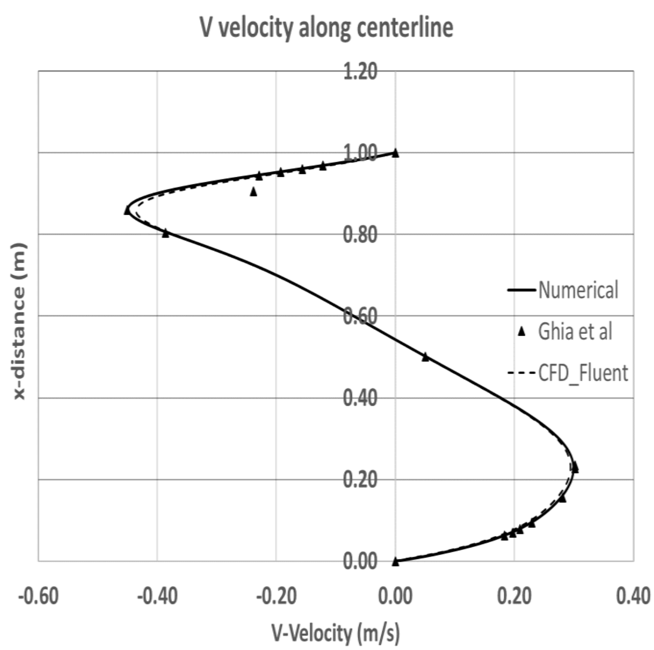
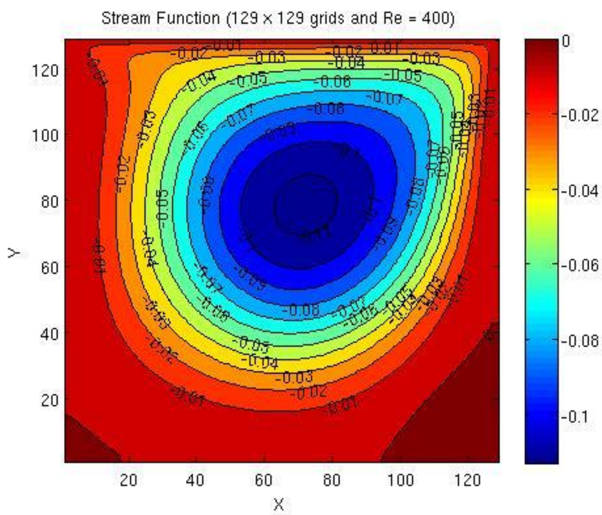
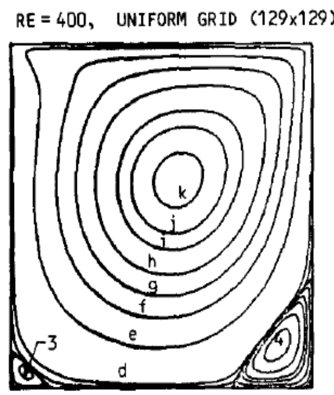


Figure 10 Comparison of V Velocity profile along y-centerline for RE=400 (Numerical Code vs Ghia et al. vs CFD analysis)





Contour letter	Value of ψ
a	-1.0×10^{-10}
b	-1.0×10^{-7}
c	-1.0×10^{-5}
d	-1.0×10^{-4}
e	-0.0100
f	-0.0300
g	-0.0500
h	-0.0700
i	-0.0900
j	-0.1000
k	-0.1100
l	-0.1150
m	-0.1175

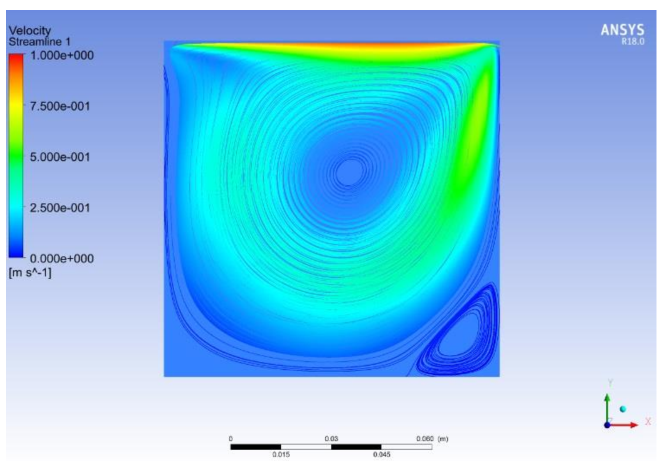
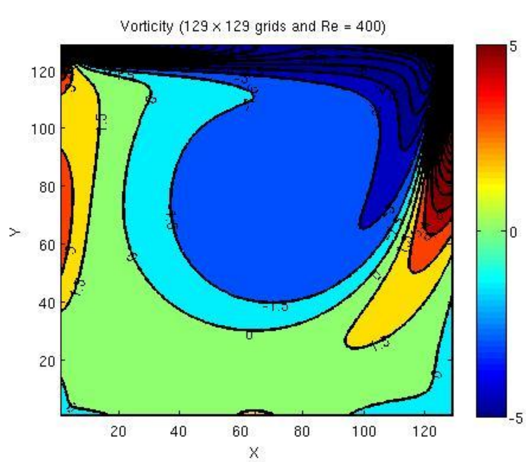
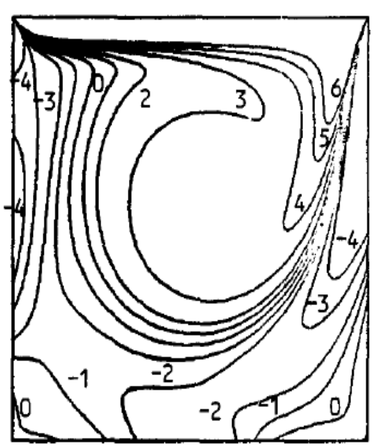


Figure 11 Comparison of Streamlines pattern for RE=400 (Numerical Code vs Ghia et al. vs CFD analysis)





Contour number	Value of ω
0	0.0
± 1	± 0.5
± 2	± 1.0
± 3	± 2.0
± 4	± 3.0
5	4.0
6	5.0

Figure 12 Comparison of Vorticity contours for RE=400 (Numerical Code vs Ghia et al)

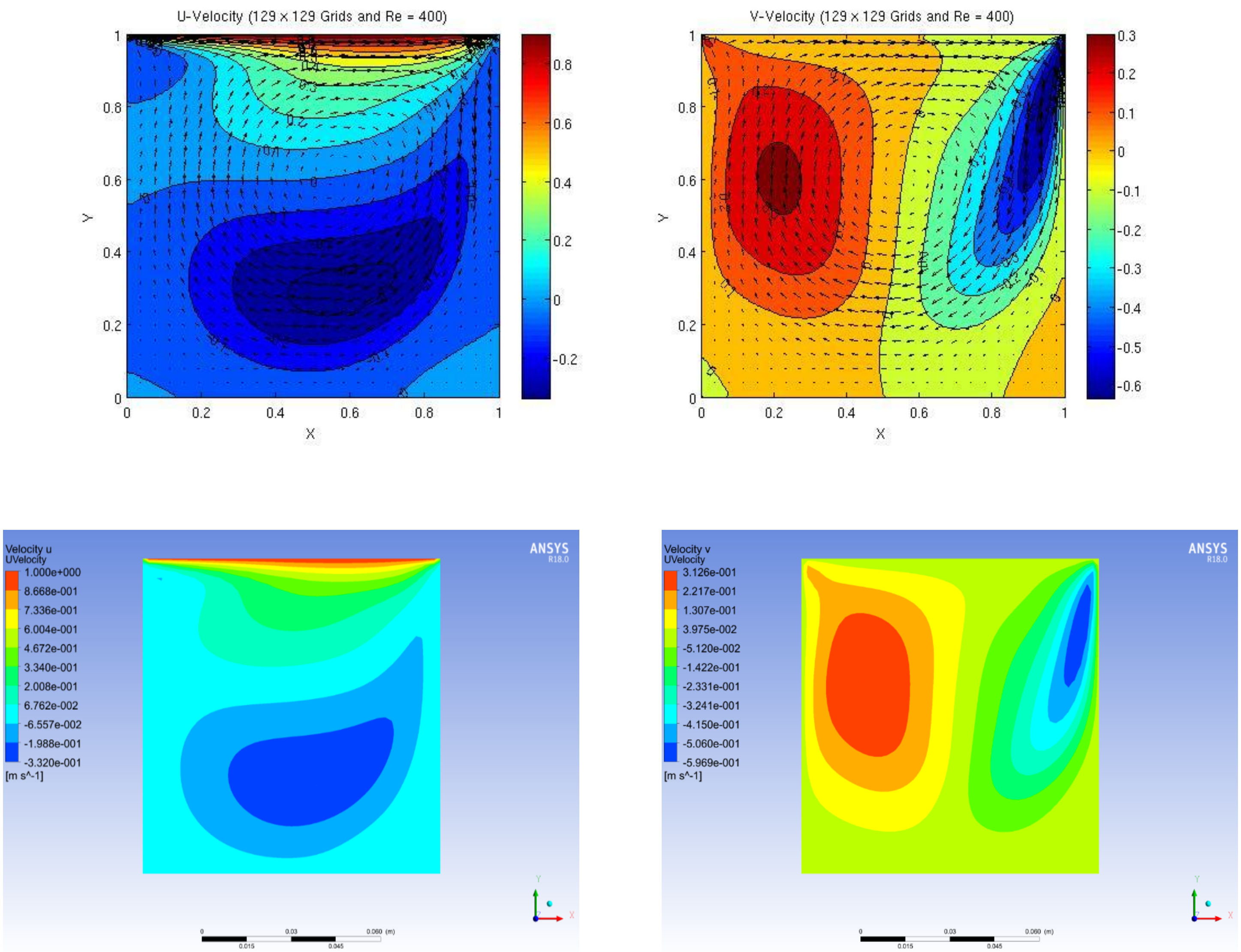
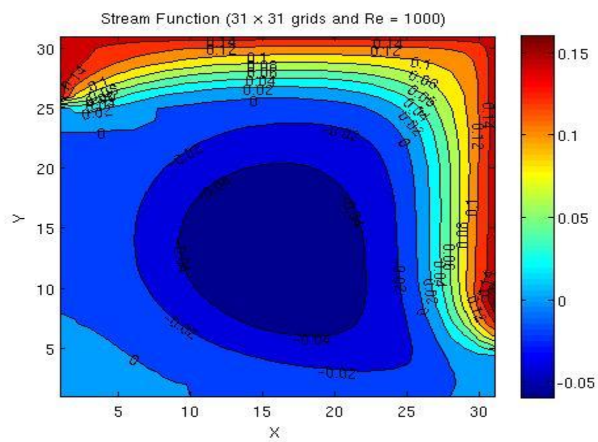


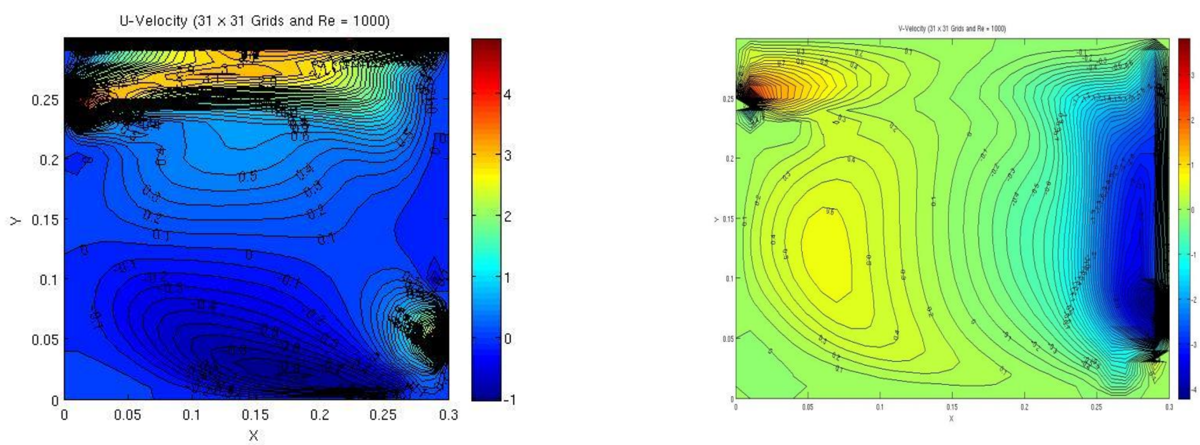
Figure 13 Comparison of U & V velocity contours for RE=400 (Numerical Code vs CFD analysis)

6.2 Lid Driven Cavity with inlet & outlet ports (Results for RE=1000):

The results for Lid driven cavity with inlet and outlet ports are presented in Figure 14 and 15. The streamline patterns in the Figure 14 (a) are in good qualitative agreement in comparison stream lines pattern presented in *Computational Fluid Dynamics Vol I by Klaus A. Hoffmann and Steve T. Chiang in pg.no 351.* Figure 15 represents U and V component velocities along centerline for Re =1000.



(a)



(b)

Figure 14 Stream Function, U and V Contours for Lid Driven with Open flow channel for Re = 1000

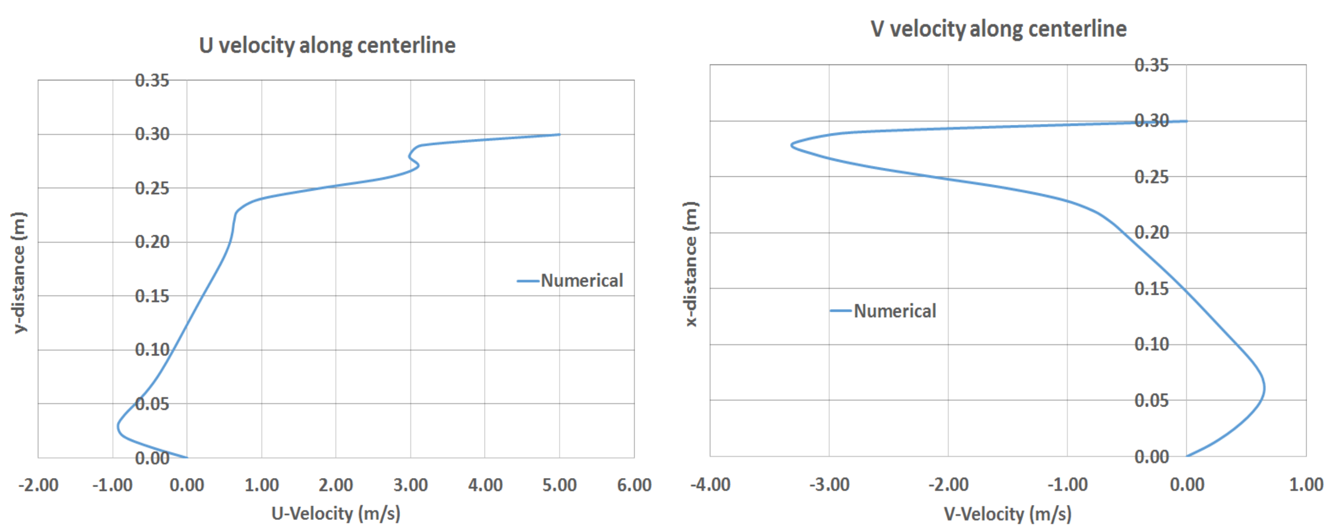


Figure 15 U &V velocities along centerline for RE=1000 (Numerical Code)

7 Appendix

7.1 Fortran Code (Lid driven closed cavity)

```
!*******************************************************************************
  SUBROUTINE  READ_INPUTS (L, H, NX, NY, J1, J2, J3, J4, DT, T, UP, RE, DX, DY, X, Y, U, V, SF, W, P, RSF, RW, PRHS)
  IMPLICIT NONE
  DOUBLE PRECISION :: L, H, DT, T, UP, RE
  INTEGER :: NX, NY, J1, J2, J3, J4
  DOUBLE PRECISION :: DX,DY
  DOUBLE PRECISION, ALLOCATABLE :: X(:),Y(:)
  DOUBLE PRECISION, ALLOCATABLE :: U(:,:), V(:,:), SF(:,:), W(:,:), RSF(:,:), RW(:,:), P(:,:), PRHS(:,:)
  INTEGER        :: I,J

  CHARACTER*1000  ::  DUMMYLINE

  OPEN (UNIT=01,FILE='AM16D025.In',STATUS='OLD')

  READ (01,*)         DUMMYLINE, L
  READ (01,*)         DUMMYLINE, H
  READ (01,*)         DUMMYLINE, NX
  READ (01,*)         DUMMYLINE, NY
  READ (01,*)         DUMMYLINE, J1
  READ (01,*)         DUMMYLINE, J2
  READ (01,*)         DUMMYLINE, J3
  READ (01,*)         DUMMYLINE, J4
  READ (01,*)         DUMMYLINE, DT
  READ (01,*)         DUMMYLINE, T
  READ (01,*)         DUMMYLINE, UP
  READ (01,*)         DUMMYLINE, RE

  ALLOCATE (X(NX))
  ALLOCATE (Y(NY))
  ALLOCATE (U(NX,NY))
  ALLOCATE (V(NX,NY))
  ALLOCATE (SF(NX,NY))
  ALLOCATE (W(NX,NY))
  ALLOCATE (RSF(NX,NY))
  ALLOCATE (RW(NX,NY))
  ALLOCATE (P(NX,NY))
  ALLOCATE (PRHS(NX,NY))

  DX = L/(NX-1)
  DY = H/(NY-1)
  DO I=1,NX
     X(I) = (I-1)*DX
  ENDDO
  DO J=1,NY
     Y(J) = (J-1)*DY
  ENDDO
  U = 0.0d0
  V = 0.0d0
  SF = 0.0d0
  W = 0.0d0
  RW = 0.0d0
  RSF = 0.0d0
  PRHS = 0.0d0
  P = 0.0d0
  DO I=1,NX
    U(I,NY) = UP
  ENDDO

  END SUBROUTINE
!*******************************************************************************************************************
  PROGRAM LID_DRIVEN_CAVITY_STREAM_FUNCTION_VORTICITY_APPROACH
  IMPLICIT NONE

  DOUBLE PRECISION :: L, H, DT, T, UP, RE
  INTEGER :: NX, NY, J1, J2, J3, J4, K
  DOUBLE PRECISION :: DX,DY
```

```
DOUBLE PRECISION, ALLOCATABLE :: X(:),Y(:)
DOUBLE PRECISION, ALLOCATABLE :: U(:,:), V(:,:), SF(:,:), W(:,:), P(:,:), RSF(:,:), RW(:,:), SF_OLD(:,:), PRHS(:,:), PNEW(:,:), W_OLD(:,:)
INTEGER         :: I,J, NUM_TIMESTEPS, ITER, B, A, C
DOUBLE PRECISION    :: ERROR, ITC, ERR, ERR1
DOUBLE PRECISION, PARAMETER    :: ESF = 1E-5
DOUBLE PRECISION, PARAMETER    :: RF = 1.5
DOUBLE PRECISION :: TERM1, TERM2, TERM3, TERM4, SF_CONST,ERRW, ERRSF
CHARACTER*1000   ::  XX,Z
character(len=8) :: fmt ! format descriptor

REAL :: START, FINISH
SF_CONST = 0;

CALL READ_INPUTS (L, H, NX, NY, J1, J2, J3, J4, DT, T, UP, RE, DX, DY, X, Y, U, V, SF, W, P, RSF, RW, PRHS)

ALLOCATE (SF_OLD(NX,NY))
        ALLOCATE (W_OLD(NX,NY))
ALLOCATE (PNEW(NX,NY))

NUM_TIMESTEPS = T/DT
ERROR = 100.0d0
ITC = 0
CALL CPU_TIME(START)
DO WHILE((ITC.LT.NUM_TIMESTEPS))
                        W_OLD = W
  DO I=2,NX-1
    DO J=2,NY-1
      TERM1 = (W(I+1,J)-2.0D0*W(I,J)+W(I-1,J))/(DX*DX)
      TERM2 = (W(I,J+1)-2.0D0*W(I,J)+W(I,J-1))/(DY*DY)
      TERM3 = (U(I+1,J)*W(I+1,J)-U(I-1,J)*W(I-1,J))/(2.0D0*DX)
      TERM4 = (V(I,J+1)*W(I,J+1)-V(I,J-1)*W(I,J-1))/(2.0D0*DY)
      RW(I,J) = (TERM1+TERM2)/RE-TERM3-TERM4
      W(I,J) = W(I,J)+DT*RW(I,J)
    ENDDO
  ENDDO
  DO ITER = 1, 1000
    SF_OLD=SF
    DO I = 2 , NX-1
      DO J  = 2 , NY-1
        SF (I,J) = (RF/4)*((W(I,J)*(DX*DX))+SF(I+1,J)+SF(I,J+1)+SF(I-1,J)+SF(I,J-1))+(1-RF)*SF(I,J)
      ENDDO
    ENDDO
    ERR = 0.0
    DO I = 1 , NX
      DO J  = 1 , NY
        ERR = ERR+ABS(SF_OLD(I,J)-SF(I,J))
      ENDDO
    ENDDO
    IF (ERR .LE. 0.001) THEN
      EXIT
    ENDIF
  ENDDO

  DO J=2,NY-1
    SF(2,J) = 0.25d0*SF(3,J)
    SF(NX-1,J) = 0.25d0*SF(NX-2,J)
  ENDDO
  DO I=2,NX-1
    SF(I,2) = 0.25d0*SF(I,3)
    SF(I,NY-1) = 0.25d0*(SF(I,NY-2)-2.0d0*DY)
  ENDDO

  DO I = 1 , NX
    W (I,1) = (SF_CONST-2*SF(I,2))/(DY*DY)  !BOTTOMWALL
    W (I,NY) = -2*((SF(I,NY-1)-SF_CONST)/(DY*DY)+(UP/DY))   !TOPWALL
  ENDDO

  DO J = 2 , J1
    W (1,J) = 2*(SF_CONST-SF(2,J))/(DX*DX)
  ENDDO
  DO J = J1+1 , J2
    W (1,J) = 2*(SF_CONST-SF(2,J))/(DX*DX)
  ENDDO
```

```
    DO J = J2+1 , NY-1
       W (1,J) = 2*(SF_CONST-SF(2,J))/(DX*DX)
    ENDDO
    !RIGHTWALL
    DO J = 2 , J3
       W (NX,J) = 2*(SF_CONST-SF(NX-1,J))/(DX*DX)
    ENDDO
    DO J = J3+1 , J4
       W (NX,J) = 2*(SF_CONST-SF(NX-1,J))/(DX*DX)
    ENDDO
    DO J = J4+1 , NY-1
       W (NX,J) = 2*(SF_CONST-SF(NX-1,J))/(DX*DX)
    ENDDO

    DO J=1,NY
       U(1,J) = 0.0d0
       U(NX,J) = 0.0d0
       V(1,J) = 0.0d0
       V(NX,J) = 0.0d0
    ENDDO
    DO I=2,NX-1
       U(I,1) = 0.0d0
       V(I,1) = 0.0d0
       U(I,NY) = UP
       V(I,NY) = 0.0d0
    ENDDO

    DO I=2,NX-1
       DO J=2,NY-1
          U(I,J) = 0.5d0*(SF(I,J+1)-SF(I,J-1))/DY
          V(I,J) = -0.5d0*(SF(I+1,J)-SF(I-1,J))/DX
       ENDDO
    ENDDO

    DO I=2,NX-1
       PNEW = P
       DO J=2,NY-1
          PRHS(I,J) = (((SF(I-1,J)-2*SF(I,J)+SF(I+1,J))/(DX*DX))*((SF(I,J-1)-2*SF(I,J)+SF(I,J+1))/(DY*DY)))- (SF(I+1,J+1)-SF(I+1,J-1)-SF(I-1,J+1)+SF(I-1,J-
1))/(4*(DX*DY))
          P(I,J) = (0.25*(PNEW(I+1,J)+PNEW(I-1,J) + PNEW(I,J+1)+PNEW(I,J-1))- 0.5*((PRHS(I,J)*DX**2*DY**2)));
       ENDDO
    ENDDO

    ITC = ITC+1
                      ERR1 = 0.0
    DO I = 1 , NX
       DO J  = 1 , NY
          ERR1 = ERR1+ABS(W_OLD(I,J)-W(I,J))
       ENDDO
    ENDDO
    IF ((ERR1 .LE. 0.002) .AND. (ITC .GE. 5) )THEN
       EXIT
    ENDIF

 ENDDO

 CALL CPU_TIME(FINISH)
 OPEN (UNIT=10,FILE='TIME.DAT',STATUS='UNKNOWN')
 OPEN (UNIT=2,FILE='VX.DAT',STATUS='UNKNOWN')
 OPEN (UNIT=3,FILE='UY.DAT',STATUS='UNKNOWN')
 OPEN (UNIT=4,FILE='SF.DAT',STATUS='UNKNOWN')
 OPEN (UNIT=5,FILE='W.DAT',STATUS='UNKNOWN')
 OPEN (UNIT=6,FILE='U.DAT',STATUS='UNKNOWN')
 OPEN (UNIT=7,FILE='V.DAT',STATUS='UNKNOWN')
 OPEN (UNIT=8,FILE='P.DAT',STATUS='UNKNOWN')
 OPEN (UNIT=9,FILE='T.DAT',STATUS='UNKNOWN')
 DO I = 1 , NX
    WRITE (2,*) X(I), V(I,(NY+1)/2)
 END DO
 DO J = 1 , NY
    WRITE (3,*) Y(J), U((NX+1)/2,J)
 END DO
 DO I=1,NX
```

```
    DO J=1,NY
      WRITE (4,*) X(I), X(J), SF(I,J)
      WRITE (5,*) X(I), X(J), W(I,J)
      WRITE (6,*) X(I), X(J), U(I,J)
      WRITE (7,*) X(I), X(J), V(I,J)
      WRITE (8,*) X(I), X(J), P(I,J)
    ENDDO
  ENDDO
  WRITE (10,'(A, 5X, E13.3)') 'TIME.TXT', FINISH-START

  DO I = 1 , 560
    CLOSE (I)
  ENDDO


  END PROGRAM LID_DRIVEN_CAVITY_STREAM_FUNCTION_VORTICITY_APPROACH
```

7.2 Fortran Code (lid driven cavity with inlet & outlet ports)

```
!**********************************************************************
     SUBROUTINE  READ_INPUTS (L, H, NX, NY, J1, J2, J3, J4, DT, T, UP, RE, DX, DY, X, Y, U, V, SF, W, P, RSF,
RW, PRHS)
    IMPLICIT NONE
    DOUBLE PRECISION :: L, H, DT, T, UP, RE
    INTEGER :: NX, NY, J1, J2, J3, J4
    DOUBLE PRECISION :: DX,DY
    DOUBLE PRECISION, ALLOCATABLE :: X(:),Y(:)
    DOUBLE PRECISION, ALLOCATABLE :: U(:,:), V(:,:), SF(:,:), W(:,:), RSF(:,:), RW(:,:), P(:,:), PRHS(:,:)
    INTEGER          :: I,J

    CHARACTER*1000   ::  DUMMYLINE

    OPEN (UNIT=999,FILE='AM16D025.In',STATUS='OLD')

    READ (999,*)            DUMMYLINE, L
    READ (999,*)            DUMMYLINE, H
    READ (999,*)            DUMMYLINE, NX
    READ (999,*)            DUMMYLINE, NY
    READ (999,*)            DUMMYLINE, J1
    READ (999,*)            DUMMYLINE, J2
    READ (999,*)            DUMMYLINE, J3
    READ (999,*)            DUMMYLINE, J4
    READ (999,*)            DUMMYLINE, DT
    READ (999,*)            DUMMYLINE, T
    READ (999,*)            DUMMYLINE, UP
    READ (999,*)            DUMMYLINE, RE

    ALLOCATE (X(NX))
    ALLOCATE (Y(NY))
    ALLOCATE (U(NX,NY))
    ALLOCATE (V(NX,NY))
    ALLOCATE (SF(NX,NY))
    ALLOCATE (W(NX,NY))
    ALLOCATE (RSF(NX,NY))
    ALLOCATE (RW(NX,NY))
    ALLOCATE (P(NX,NY))
    ALLOCATE (PRHS(NX,NY))

    DX = L/(NX-1)
    DY = H/(NY-1)
    DO I=1,NX
        X(I) = (I-1)*DX
    ENDDO
    DO J=1,NY
        Y(J) = (J-1)*DY
    ENDDO
    U = 0.0d0
    V = 0.0d0
    SF = 0.0d0
    W = 0.0d0
    RW = 0.0d0
    RSF = 0.0d0
    PRHS = 0.0d0
```

```
    P = 0.0d0
    DO I=1,NX
       U(I,NY) = UP
    ENDDO

    END SUBROUTINE
!*************************************************************************************************************
*********
    PROGRAM LID_DRIVEN_CAVITY_STREAM_FUNCTION_VORTICITY_APPROACH
    IMPLICIT NONE

    DOUBLE PRECISION :: L, H, DT, T, UP, RE
    INTEGER :: NX, NY, J1, J2, J3, J4, K
    DOUBLE PRECISION :: DX,DY, SF1, SF2
    DOUBLE PRECISION, ALLOCATABLE :: X(:),Y(:)
    DOUBLE PRECISION, ALLOCATABLE :: U(:,:), V(:,:), SF(:,:), W(:,:), P(:,:), RSF(:,:), RW(:,:), SF_OLD(:,:),
PRHS(:,:), PNEW(:,:), W_OLD(:,:)
    INTEGER          :: I,J, NUM_TIMESTEPS, ITER, B, A, C
    DOUBLE PRECISION    :: ERROR, ITC, ERR, ERR1
    DOUBLE PRECISION, PARAMETER    :: ESF = 1E-5
    DOUBLE PRECISION, PARAMETER    :: RF = 1.5
    DOUBLE PRECISION :: TERM1, TERM2, TERM3, TERM4, SF_CONST,ERRW, ERRSF
    CHARACTER*1000   ::  XX,Z
    character(len=8) :: fmt ! format descriptor

    REAL :: START, FINISH
    SF_CONST = 0;

    SF1 = 0.0;
    SF2 = 0.15;


    CALL READ_INPUTS (L, H, NX, NY, J1, J2, J3, J4, DT, T, UP, RE, DX, DY, X, Y, U, V, SF, W, P, RSF, RW, PRHS)

    ALLOCATE (SF_OLD(NX,NY))
    ALLOCATE (W_OLD(NX,NY))
    ALLOCATE (PNEW(NX,NY))

    DO I=1,NX
        U(I,NY) = UP
    ENDDO

    NUM_TIMESTEPS = T/DT
    ITC = 0
    CALL CPU_TIME(START)
    DO WHILE((ITC.LT.NUM_TIMESTEPS))
        W_OLD = W
        !--------------------------------START:STREAM FUNCTION BC's-------------------------------------------
---
       !LEFTWALL
        DO J = 1 , J1-1
            !SF(2,J) = 0.25d0*(SF(3,J) + 3*SF1)
            !SF(1,J) = 1/3 * (4*SF(2,J)-SF(3,J))
            SF(1,J) = SF1
        ENDDO
        DO J = J1 , J2
            SF(1,J) = 1/3 * (4*SF(2,J)-SF(3,J))
        ENDDO
        DO J = J2+1 , NY
            !SF(2,J) = 0.25d0*(SF(3,J) + 3*SF2)
            SF(1,J) = SF2
        ENDDO

        !RIGHTWALL
        DO J = 1 , J3-1
            !SF(NX-1,J) = 0.25d0*(SF(NX-2,J)+3*SF1)
            SF(NX,J) = SF1
        ENDDO
        DO J = J3 , J4
            SF(NX,J) = - SF(NX-2,J) + 2*SF(NX-1,J)
        ENDDO
        DO J = J4+1 , NY
           !SF(NX-1,J) = 0.25d0*(SF(NX-2,J)+3*SF2)
           SF(NX,J) = SF2
        ENDDO
```

```
DO I=2,NX-1
    SF(I,1) = SF1 !BOTTOM WALL
    SF(I,NY) = SF2 !TOP WALL
ENDDO
!---------------------------------END:STREAM FUNCTION BC's---------------------------------------------

DO ITER = 1, 1000
    SF_OLD=SF
    DO I = 2 , NX-1
        DO J  = 2 , NY-1
            SF (I,J) = (RF/4)*((W(I,J)*(DX*DX))+SF(I+1,J)+SF(I,J+1)+SF(I-1,J)+SF(I,J-1))+(1-RF)*SF(I,J)
        ENDDO
    ENDDO
    ERR = 0.0
    DO I = 1 , NX
        DO J  = 1 , NY
            ERR = ERR+ABS(SF_OLD(I,J)-SF(I,J))
        ENDDO
    ENDDO
    IF (ERR .LE. 0.001) THEN
        EXIT
    ENDIF
ENDDO

!---------------------------------START:VORTICITY BC's-----------------------------------------------
DO I = 2 , NX-1
    W (I,1) = 2*(SF(I,1)-SF(I,2))/(DY*DY)  !BOTTOMWALL
    W (I,NY) = -2*((SF(I,NY-1)-SF(I,NY))/(DY*DY)+(UP/DY))   !TOPWALL
ENDDO

DO J = 1 , J1-1
    W (1,J) = 2*(SF(1,J)-SF(2,J))/(DX*DX)
ENDDO
DO J = J1 , J2
    W (1,J) = (2*(SF(1,J)-SF(2,J))/(DX*DX)) - (SF(1,J+1)-2*SF(1,J) + SF(1,J-1))/(DY*DY)
ENDDO
DO J = J2+1 , NY
    W (1,J) = 2*(SF(1,J)-SF(2,J))/(DX*DX)
ENDDO
!RIGHTWALL
DO J = 1 , J3
    W (NX,J) = 2*(SF(NX,J)-SF(NX-1,J))/(DX*DX)
ENDDO
DO J = J3 , J4
    W (NX,J) = 1/3 * (4 *SF(NX-1,J)-SF(NX-2,J))
ENDDO
DO J = J4+1 , NY
    W (NX,J) = 2*(SF(NX,J)-SF(NX-1,J))/(DX*DX)
ENDDO
!---------------------------------END:VORTICITY BC's---------------------------------------------

DO I=2,NX-1
    DO J=2,NY-1
        U(I,J) = 0.5d0*(SF(I,J+1)-SF(I,J-1))/DY
        V(I,J) = -0.5d0*(SF(I+1,J)-SF(I-1,J))/DX
    ENDDO
ENDDO

DO I=2,NX-1
    DO J=2,NY-1
        TERM1 = (W(I+1,J)-2.0D0*W(I,J)+W(I-1,J))/(DX*DX)
        TERM2 = (W(I,J+1)-2.0D0*W(I,J)+W(I,J-1))/(DY*DY)
        TERM3 = (U(I+1,J)*W(I+1,J)-U(I-1,J)*W(I-1,J))/(2.0D0*DX)
        TERM4 = (V(I,J+1)*W(I,J+1)-V(I,J-1)*W(I,J-1))/(2.0D0*DY)
        RW(I,J) = (TERM1+TERM2)/RE-TERM3-TERM4
        W(I,J) = W(I,J)+DT*RW(I,J)
    ENDDO
ENDDO


DO I=2,NX-1
    PNEW = P
    DO J=2,NY-1
```

```
                PRHS(I,J) = (((SF(I-1,J)-2*SF(I,J)+SF(I+1,J))/(DX*DX))*((SF(I,J-1)-
2*SF(I,J)+SF(I,J+1))/(DY*DY)))- (SF(I+1,J+1)-SF(I+1,J-1)-SF(I-1,J+1)+SF(I-1,J-1))/(4*(DX*DY))
                P(I,J) = (0.25*(PNEW(I+1,J)+PNEW(I-1,J) + PNEW(I,J+1)+PNEW(I,J-1))-
0.5*((PRHS(I,J)*DX**2*DY**2)));
            ENDDO
        ENDDO

        ITC = ITC+1
        ERR1 = 0.0
        DO I = 1 , NX
            DO J  = 1 , NY
                ERR1 = ERR1+ABS(W_OLD(I,J)-W(I,J))
            ENDDO
        ENDDO
        IF ((ERR1 .LE. 0.002) .AND. (ITC .GE. 5) )THEN
            EXIT
        ENDIF
    ENDDO

    CALL CPU_TIME(FINISH)
    OPEN (UNIT=10,FILE='TIME.DAT',STATUS='UNKNOWN')
    OPEN (UNIT=2,FILE='VX.DAT',STATUS='UNKNOWN')
    OPEN (UNIT=3,FILE='UY.DAT',STATUS='UNKNOWN')
    OPEN (UNIT=4,FILE='SF.DAT',STATUS='UNKNOWN')
    OPEN (UNIT=5,FILE='W.DAT',STATUS='UNKNOWN')
    OPEN (UNIT=6,FILE='U.DAT',STATUS='UNKNOWN')
    OPEN (UNIT=7,FILE='V.DAT',STATUS='UNKNOWN')
    OPEN (UNIT=8,FILE='P.DAT',STATUS='UNKNOWN')
    OPEN (UNIT=9,FILE='T.DAT',STATUS='UNKNOWN')
    DO I = 1 , NX
        WRITE (2,*) X(I), V(I,(NY+1)/2)
    END DO
    DO J = 1 , NY
        WRITE (3,*) Y(J), U((NX+1)/2,J)
    END DO
    DO I=1,NX
        DO J=1,NY
            WRITE (4,*) X(I), X(J), SF(I,J)
            WRITE (5,*) X(I), X(J), W(I,J)
            WRITE (6,*) X(I), X(J), U(I,J)
            WRITE (7,*) X(I), X(J), V(I,J)
            WRITE (8,*) X(I), X(J), P(I,J)
        ENDDO
    ENDDO
    WRITE (10,'(A, 5X, E13.3)') 'TIME.TXT', FINISH-START

    DO I= 1 , 10
        CLOSE (I)
    ENDDO


    END PROGRAM LID_DRIVEN_CAVITY_STREAM_FUNCTION_VORTICITY_APPROACH
```