Master Of Science in Computational Mechanics

Finite Elements in Fluids

Chinmay Khisti

(23 April 2019)

Assignment 7

(Unsteady Incompressible Flow)

The Navier-Stokes equations with unsteady, isothermal and incompressible fluid flow is stated as below:

$$\begin{array}{ll} v_t + (v \cdot \nabla)v - \nu \nabla^2 v + \nabla p = f & in\Omega \\ \nabla \cdot v = 0 & in\Omega \\ v = v_D & on\Gamma_D \\ n \cdot \sigma = t & on\Gamma_N \end{array} \tag{1}$$

The balance of momentum equation 1 consist of unsteady term, viscous term, convective term, pressure gradient and external body forces.
The weal form This Navier-Stokes equations is compatible with a 2D flow in a cavity (Classical problem of the lid-driven square cavity flow). Thus following boundary conditions were taken into consideration while solving the problem as it can be seen in the figure 1

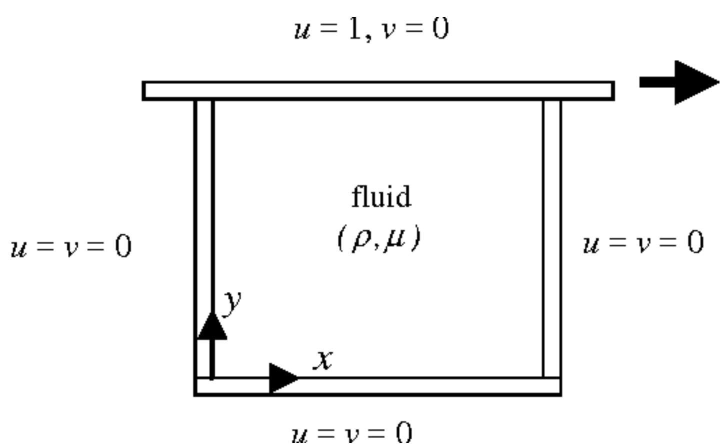


Figure 1: Boundary conditions

The equation 1 is then discretized in space using Galerkin formulation we obtain equation 2

$$\begin{array}{l} M\dot{u}(t) + [K + C(v)]u + G_p = f \\ G^T u(t) = 0 \\ u(0) = v(0) - v_D(0) \end{array} \tag{2}$$

$$M = \begin{bmatrix} N_1 & 0 \\ 0 & N_1 \\ N_2 & 0 \\ 0 & N_2 \\ . & . \\ . & . \\ . & . \\ . & . \\ N_n & 0 \\ 0 & N_n \end{bmatrix} \begin{bmatrix} N_1 & 0 & N_2 & 0 & .... & N_n & 0 \\ 0 & N_1 & 0 & N_2 & .... & 0 & N_n \end{bmatrix}$$

where $matM$ is a mass matrix. Now there are several ways to discretize in time of which following two methods were used for this problem.

1 Semi-Implicit Method

In this method for discretization purpose θ method for time discretization can be implemented as seen in equation 3

$$u_t = \frac{f - (K + c(v))u - Gp}{M}$$

the equation becomes

$$\frac{\Delta u}{\Delta t} - \theta \Delta u_t = u_t^n \tag{3}$$

where,
n = time step,
θ = (1)Implicit or (0) Explicit method.
By plugging in equation 2 in equation 3 we get following:

$$\frac{\Delta u}{\Delta t} - \theta(\frac{-(K + C^{n+1})u^{n+1} + (K + C^n)u^n - Gp^{n+1} + Gp^n}{M}) = \frac{f - (K + c^n)u^n - Gp^n}{M} \tag{4}$$

The Semi-implicit method implies them term C is considered linearized so the above equation becomes:

$$\frac{\Delta u}{\Delta t} - \theta(\frac{(K + C^n)\Delta u + G\Delta p}{M}) = \frac{f - (K + c^n)u^n - Gp^n}{M}$$

Considering divergence condition the equation becomes:

$$\begin{aligned} &(M + \theta\Delta t(K + C^n))\Delta u + \theta\Delta t G\Delta p) = \Delta t(f - (K + c^n)u^n - Gp^n) \\ &G^T \Delta u = 0 \end{aligned} \tag{5}$$

It can be rewritten in matrix form as,

$$\begin{bmatrix} M + \theta\Delta t(K + C^n) & \theta\Delta t G \\ G^T & 0 \end{bmatrix} \begin{bmatrix} \Delta u \\ \Delta t \end{bmatrix} = \begin{bmatrix} \Delta t(f - (K + c^n)u^n - Gp^n) \\ 0 \end{bmatrix} \tag{6}$$

The piece of the code implemented:

```
while stp < nstep
    stp = stp +1;
    C = ConvectionMatrix(X,T,referenceElement,velo);
    Cr = C(dofUnk,dofUnk);
    fred_n = fred - (K(dofUnk,dofDir)+C(dofUnk,dofDir))*valDir;

    Att = [Mred+teta*dt*(Kred+Cr)    dt*teta*Gred'    Gred    L];
    btt = [dt*(fred_n-(Kred+Cr)*veloVect(dofUnk)-Gred'*pres); f_q];
    %velocity and pressure increment
    soli = Att\btt;

    % Updating the solution
    veloi = zeros(ndofV,1);
    veloi(dofUnk) = soli(1:nunkV);
    presi = soli(nunkV+1:end);
    velo = velo + reshape(veloi,2,[])';
    pres = pres + presi;
end
```

2 Chorin-Temam Method

The projection method is an effective means of numerically solving time-dependent incompressible fluid-flow problems. This method calculates pressure and velocity fields separately in two step form. The equation 2 is divided into two separate parts one with viscous and convective term by neglecting incompressibility of flow and one without viscous and convective part in incompressible fluid-flow giving first step,

$$Mu_t + (K + C)u = f$$
$$(M + \Delta t(K + C^n))u^* = \Delta b + Mu^n$$

Second step, which is current time step is formulated as,

$$\begin{bmatrix} M & \Delta tG \\ G^T & 0 \end{bmatrix} \begin{bmatrix} u^{n+1} \\ p^{n+1} \end{bmatrix} = \begin{bmatrix} Mu^* \\ 0 \end{bmatrix} \tag{7}$$

The piece of the code implemented:

```
while stp < nstep
    stp = stp +1;
    C = ConvectionMatrix(X,T,referenceElement,velo);
    Cr = C(dofUnk,dofUnk);
    fred_n = fred - (K(dofUnk,dofDir)+C(dofUnk,dofDir))*valDir;
```

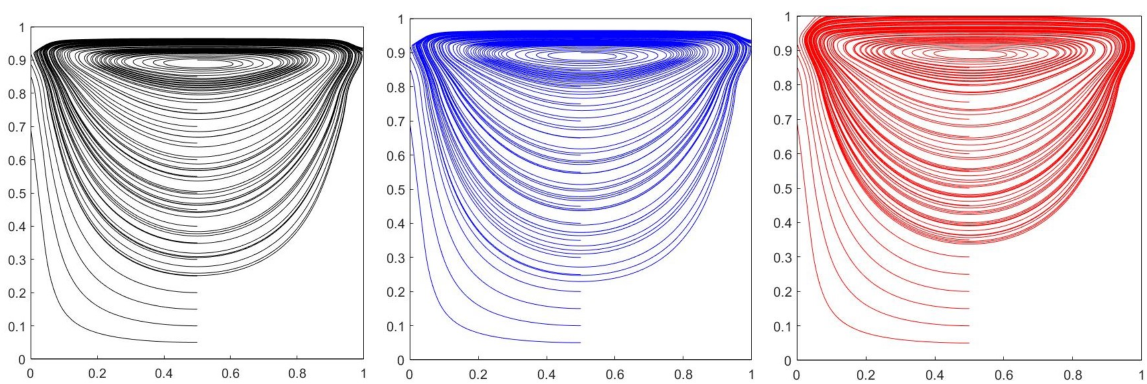
```
    % FIRST STEP
    btt = dt*fred_n+Mred*veloVect(dofUnk);
    Att = Mred+dt*(Cr+Kred);
    sol_1 = Att\btt;

    % SECOND STEP
    btt = [Mred*sol_1; f_q];
    Att = [Mred Gred'*dt; Gred L];
    soln = Att\btt;

    veloi = zeros(ndofV,1);
    veloi(dofUnk) = soln(1:nunkV);
    presi = soln(nunkV+1:end);
    velo = reshape(veloi,2,[])';
    pres = presi;
end
```

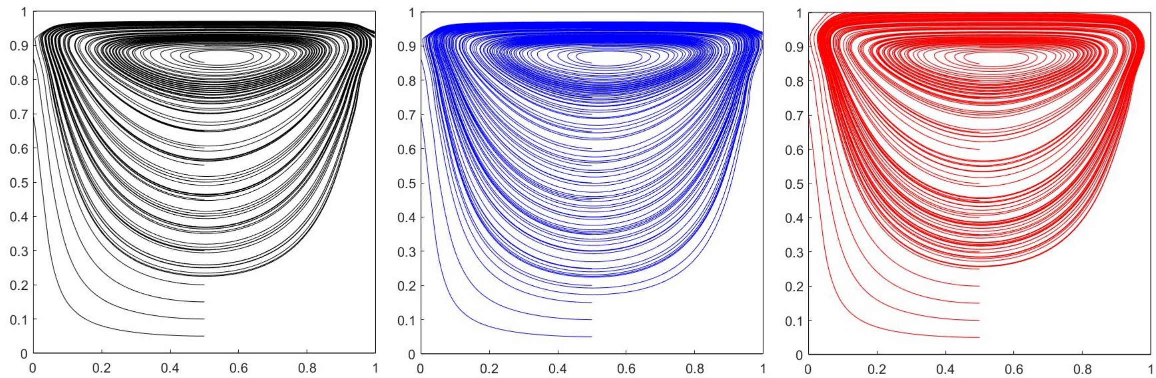
3 Results

Mesh size considered was 10x10 and element type Q2Q1. Following results were plotted for different number of time step with θ equals 1 (Black streamlines), 0.5 (Blue streamlines) and for Chorin-Temam (Red).



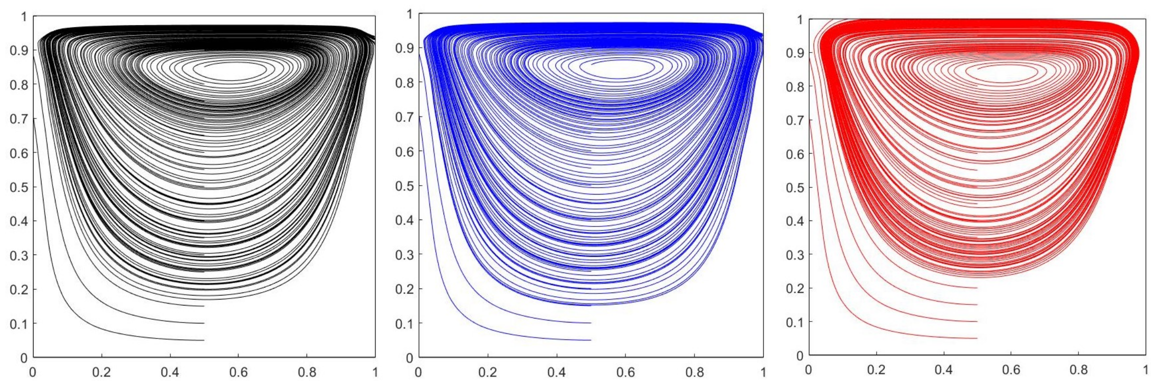
(a) $\theta = 1$ (b) $\theta = 0.5$ (c) Chorin-Temam

Figure 2: Number of time steps = 10



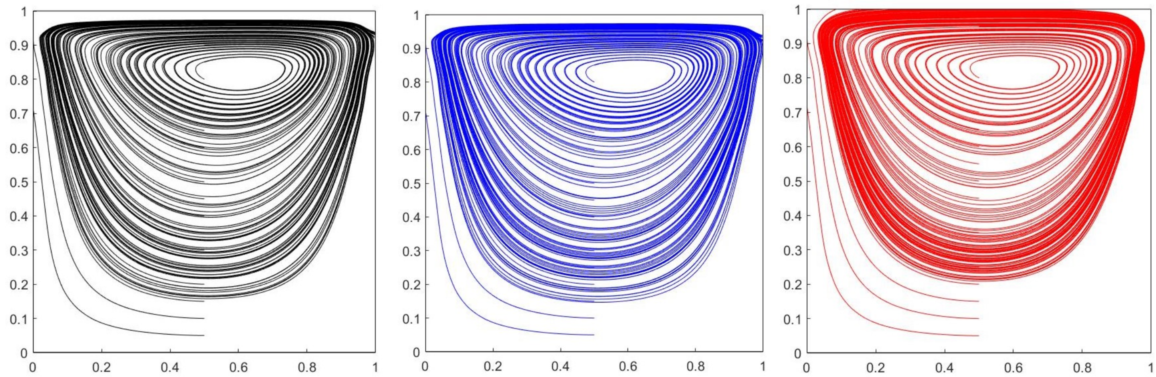
(a) $\theta = 1$ (b) $\theta = 0.5$ (c) Chorin-Temam

Figure 3: Number of time steps = 25



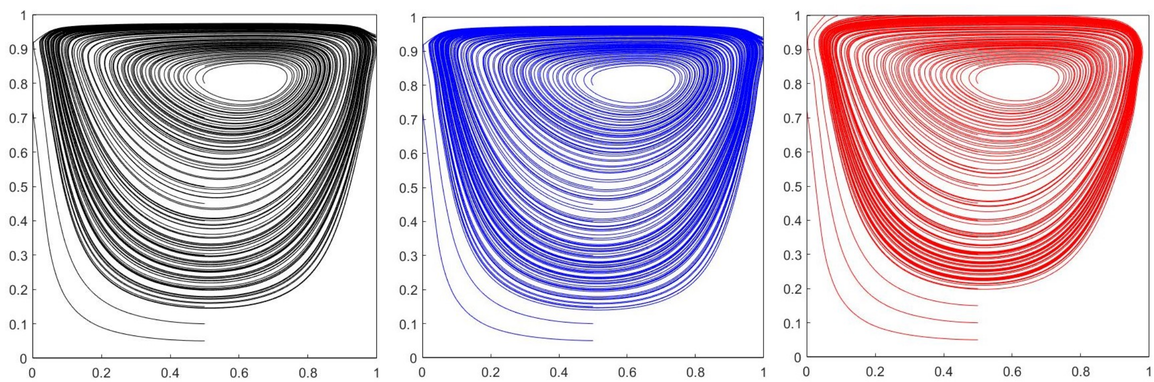
(a) $\theta = 1$ (b) $\theta = 0.5$ (c) Chorin-Temam

Figure 4: Number of time steps = 50



(a) $\theta = 1$ (b) $\theta = 0.5$ (c) Chorin-Temam

Figure 5: Number of time steps = 75



(a) $\theta = 1$ (b) $\theta = 0.5$ (c) Chorin-Temam

Figure 6: Number of time steps = 100

As we can see there is not much difference observed for various times steps for the two methods except the flow in upper side of the domain is considerably different for Semi-Implicit Method and Chorin-Temam method