Transient incompressible Navier-Stokes

Shushu Qin

April 23, 2019

1 Introduction

We have seen in the convection dominate diffusion problem that the traditional Galerkin finite element method will be unstable. In such cases, some stabilization techniques should be utilized to obtain reliable numerical solutions. Moreover, the saddle point problem we have seen in the Stokes flow also require us to satisfy the LBB condition. Specific stabilization methods are also needed to solve this problem. There are therefore two major group of methods for stabilizing the incompressible Navier-Stokes equation. Similar to last assignment of solving the steady cavity flow equation, we can add elemental stabilization terms in the weak form such as SUPG, GLS, SGS or LS. On the other hand, we can apply fractional-step approach to time integration in order to alleviate the numerical difficulties related to the saddle-point problem which arises from the variational formulation of the Navier-Stokes equations.

In this project, we implement a fractional-step method to solve the transient incompressible Navier-Stokes problem, in particular, the cavity flow problem. The basic idea is to split the numerical treatment of the various operators in the equations, thus decomposing the initially difficult problem into relatively easier substeps. There are several ways to perform such splitting. In this project, we apply Chroin-Teman projection method to perform the time integration. The overall order of the accuracy is dependent on how we approximate the time discretization. Three different approximations of the convection term of first order accuracy are considered: explicit Euler, semi implicit and implicit Euler. In the implicit method, we introduce the Newton-Raphson scheme to solve the non-linear system since it is quadratic convergent. LU decomposition is applied to solve the second step of Chorin-Temam method.

2 Chorin-Temam projection method

2.1 Strong form

In the following part, detailed maths derivation is demonstrated with the implicit Euler method. The derivation of explicit Euler and semi-implicit methods is similar. The strong form of the unsteady viscous incompressible flow we considered in this report is written as:

$$\begin{cases} \boldsymbol{v}_t + (\boldsymbol{v}\cdot\boldsymbol{\nabla})\boldsymbol{v} + \boldsymbol{\nabla} p = 0 \quad \boldsymbol{x}\in\Omega \\ \boldsymbol{\nabla}\cdot(\boldsymbol{v}) = 0 \quad \boldsymbol{x}\in\Omega \\ \boldsymbol{v} = \boldsymbol{v}_D \quad \boldsymbol{x}\in\Gamma_D \\ \boldsymbol{v}(\boldsymbol{x},0) = \boldsymbol{v}_0(\boldsymbol{x}) \quad \boldsymbol{x}\in\Omega \end{cases} \tag{1}$$

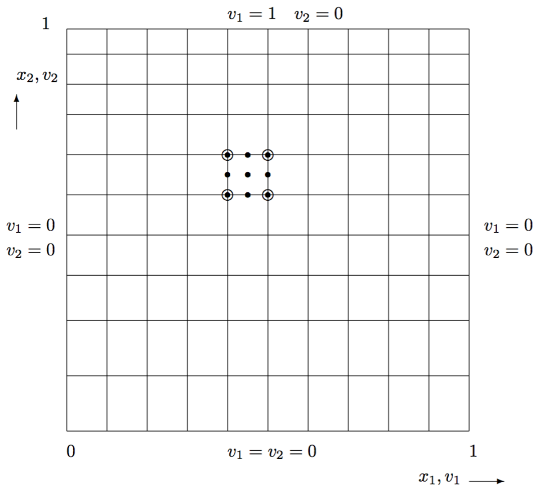


Figure 1: Boundary condition

This cavity flow example has become a standard benchmark test for incompressible flows. The boundary conditions are indicated in Figure 1. As shown in the figure, we consider the plane flow of an isothermal fluid in a square lid-driven cavity. The upper side of the cavity moves in its own plane at unit speed, while the other sides are fixed. There is no source term for this problem. Initially, the velocity is defined as 0. Then after constantly implement unit velocity at the top boundary, we will observe the flow start to rotate. The steady state solution of the cavity flow with $R_e = 100$ is shown as follows:

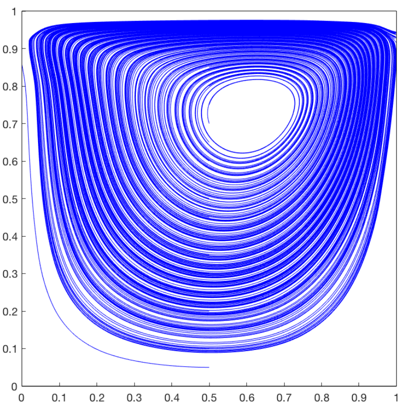


Figure 2: Streamlines of cavity flow $R_e = 100$

These equations are solved with the projection method – Chorin-Temam method. The principle of it is to compute the velocity and pressure fields separately through the computation of an intermediate velocity, namely $\boldsymbol{v}_{int}$. The Chorin-Temam projection method includes two basic steps as follows.

2.2 First step

In the first step, we solve a transient convection-diffusion problem of velocity with all Dirichlet boundary conditions.

$$\begin{cases} \dfrac{\boldsymbol{v}_{int}^{n+1} - \boldsymbol{v}^n}{\triangle t} + (\boldsymbol{v}^{n+1} \cdot \boldsymbol{\nabla})\boldsymbol{v}^{n+1} - \nu \boldsymbol{\nabla}^2 \boldsymbol{v}^{n+1} = \mathbf{0} \\ \boldsymbol{v}_{int}^{n+1} = \boldsymbol{v}_D^{n+1} \end{cases} \tag{2}$$

The Galerkin weak form is

$$\big(\boldsymbol{w}, \frac{\boldsymbol{v}_{int}^{n+1} - \boldsymbol{v}^n}{\triangle t}\big) + c\big(\boldsymbol{w}, \boldsymbol{v}^{n+1}, \boldsymbol{v}^{n+1}\big) + a\big(\boldsymbol{w}, \boldsymbol{v}^{n+1}\big) = 0 \tag{3}$$

Here we introduce Lagrange multiplier to solver the all Dirichlet boundary problem. The Equation 3 becomes

$$\begin{cases} \big(\boldsymbol{w}, \dfrac{\boldsymbol{v}_{int}^{n+1} - \boldsymbol{v}^n}{\triangle t}\big) + c\big(\boldsymbol{w}, \boldsymbol{v}^{n+1}, \boldsymbol{v}^{n+1}\big) + a\big(\boldsymbol{w}, \boldsymbol{v}^{n+1}\big) + \big\langle \boldsymbol{w}, \boldsymbol{\lambda} \big\rangle = 0 \\ \big\langle \boldsymbol{\gamma}, (\boldsymbol{v} - \boldsymbol{v}_D) \big\rangle = 0 \end{cases} \tag{4}$$

Consider now the approximation $\boldsymbol{v} \simeq \sum_i N_i \boldsymbol{v}_i$ with shape functions N_i and an interpolation for $\boldsymbol{\lambda}$ with a set of boundary functions $\{N_i^L(\boldsymbol{x})\}_{i=1}^l$, $\boldsymbol{\lambda} \simeq \sum_{i=1}^l \boldsymbol{\lambda}_i N_i$ for $\boldsymbol{x} \in \Gamma_D$.
The discretization of Equation 12 leads to the system of equations

$$\begin{bmatrix} \mathbf{M}/\triangle t + \mathbf{K} + \mathbf{C}(\boldsymbol{v}) & \mathbf{A}^T \\ \mathbf{A} & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \mathbf{v}_{int}^{n+1} \\ \boldsymbol{\lambda} \end{Bmatrix} = \begin{Bmatrix} \mathbf{M}/\triangle t \boldsymbol{v}^n \\ \mathbf{b}_1 \end{Bmatrix} \tag{5}$$

The definition of $\mathbf{M}$, $\mathbf{K}$ and $\mathbf{C}$ is the same as we have seen in Stokes and steady Navier-Stokes problems. The interpolation space for the Lagrange multiplier is chosen to be $N_i^L(\boldsymbol{x}) = \delta(\boldsymbol{x} - \boldsymbol{x}_i^L)$, where $\{\boldsymbol{x}_i^L\}_{i=1}^l$ is a set of points along Γ_D and is the Dirac delta function. In this case, the second equation in Equation 12 actually gives us

$$\boldsymbol{v}(\boldsymbol{x}_i^L) = \boldsymbol{v}_D(\boldsymbol{x}_i^L) \quad \text{for} \quad i = 1, ..., l \tag{6}$$

That is, $\mathbf{A}_{ij} = N_j(x_i^L)$, $\mathbf{b}_{1i} = \boldsymbol{v}_D(x_i^L)$.
For the semi-implicit method, it can be solved in one step by calculating $\mathbf{C}$ with the velocity at the former step.

$$\begin{bmatrix} \mathbf{M}/\triangle t + \mathbf{K} + \mathbf{C}(\boldsymbol{v}^n) & \mathbf{A}^T \\ \mathbf{A} & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \mathbf{v}_{int}^{n+1} \\ \boldsymbol{\lambda} \end{Bmatrix} = \begin{Bmatrix} \mathbf{M}/\triangle t \boldsymbol{v}^n \\ \mathbf{b}_1 \end{Bmatrix} \tag{7}$$

```
% Convection matrix
Cv = CreConv(velo,X,T,elemV);
btot = [dt*f+M*reshape(velo',nunk,1) ; btot1st];
Atot = [M+dt*(Cv+K)        Accd1'; ...
         Accd1             zeros(nDir1,nDir1)];
aux1 = Atot\btot;
```

For the fully-implicit method, this system of equations is

$$\begin{bmatrix} \mathbf{M}/\triangle t + \mathbf{K} + \mathbf{C}(\boldsymbol{v}_{int}^{n+1}) & \mathbf{A}^T \\ \mathbf{A} & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \mathbf{v}_{int}^{n+1} \\ \boldsymbol{\lambda} \end{Bmatrix} = \begin{Bmatrix} \mathbf{M}/\triangle t \boldsymbol{v}^n \\ \mathbf{b}_1 \end{Bmatrix} \tag{8}$$

It is solved by Newton-Raphson method as explained in the last homework.

```
        for iter = 1:20
            Cv = CreConv(velo0,X,T,elemV);
            referenceElement = SetReferenceElementStokes(elemV,degreeV,elemP,degreeP);
            dCU = dCdU_U(X,T,referenceElement,velo0);
            Atot = [M+dt*(Cv+K)      Accd1'; ...
                    Accd1            zeros(nDir1,nDir1)];
            Jtot = [M+dt*(Cv+K+dCU)      Accd1'; ...
                    Accd1            zeros(nDir1,nDir1)];
            % Computation of residual
            Rtot = btot-Atot*velotot;
            % Computation of velocity increment
            veloInc = Jtot\Rtot;
            % Update the solution
            velotot = velotot + veloInc;
            veloInc1 = reshape(veloInc(1:nunk),2,[])';
            velo0 = velo0+veloInc1;

            % Check convergence
            delta1 = max(abs(veloInc(1:nunk)));
            delta2 = max(abs(Rtot(1:nunk)));
            fprintf('Velocity increment=%8.6e, Residue max=%8.6e\n',delta1,delta2);
            deltaV = [deltaV delta1];
            residue = [residue delta2];
            if delta1 < tol*max(max(abs(velo))) && delta2 < tol
                fprintf('\nConvergence achieved in iteration number %g\n',iter);
                break
            end
        end
        aux1 = velotot;
```

For the explicit method, the matrix form is

$$\begin{bmatrix} \mathbf{M}/\triangle t & \mathbf{A}^T \\ \mathbf{A} & \mathbf{0} \end{bmatrix} \left\{ \begin{array}{c} \mathbf{v}_{int}^{n+1} \\ \boldsymbol{\lambda} \end{array} \right\} = \left\{ \begin{array}{c} (\mathbf{M}/\triangle t - \mathbf{K} - \mathbf{C}(\boldsymbol{v}))\boldsymbol{v}^n \\ \mathbf{b}_1 \end{array} \right\} \tag{9}$$

and LU decomposition is applied to solve the system of equations

```
        Atot = [M        Accd1'; ...
                Accd1    zeros(nDir1,nDir1)];
        [L1st,U1st] = lu(Atot);
        % Convection matrix
        Cv = CreConv(velo,X,T,elemV);
        btot =[dt*f+(M-dt*(Cv+K))*reshape(velo',nunk,1) ; btot1st];
        aux1 = U1st\(L1st\btot);
```

2.3 Second step

The second step of the Chorin-Temam method determines the end-of-step velocity $\boldsymbol{v}^{n+1}$ and pressure p^{n+1} solving

$$\begin{cases} \dfrac{\boldsymbol{v}^{n+1} - \boldsymbol{v}_{int}^{n+1}}{\triangle t} + \boldsymbol{\nabla} p^{n+1} = 0 \quad \boldsymbol{x} \in \Omega \\ \boldsymbol{\nabla} \cdot \boldsymbol{v}^{n+1} = 0 \quad \boldsymbol{x} \in \Omega \\ \boldsymbol{n} \cdot \boldsymbol{v}^{n+1} = \boldsymbol{n} \cdot \boldsymbol{v}_D^{n+1} \quad \boldsymbol{x} \in \Gamma \end{cases} \tag{10}$$

Now, the boundary condition only prescribes the normal component of the velocity, which can be translated as $v_x = 0$ on the left and right boundaries and $v_y = 0$ on the top and bottom boundaries. The Galerkin weak form is

$$\begin{cases} \left(\boldsymbol{w}, \dfrac{\boldsymbol{v}^{n+1} - \boldsymbol{v}_{int}^{n+1}}{\triangle t}\right) + b\left(\boldsymbol{w}, \boldsymbol{p}^{n+1}\right) = 0 \\ \qquad\qquad\qquad\qquad\qquad b\left(\boldsymbol{v}^{n+1}, q\right) = 0 \end{cases} \tag{11}$$

Similarly we introduce Lagrange multiplier. Hence, the Equation 11 becomes

$$\begin{cases} \left(\boldsymbol{w}, \dfrac{\boldsymbol{v}^{n+1} - \boldsymbol{v}_{int}^{n+1}}{\triangle t}\right) + b\left(\boldsymbol{w}, \boldsymbol{p}^{n+1}\right) + \left\langle \boldsymbol{w}, \boldsymbol{\lambda} \right\rangle = 0 \\ \left\langle \boldsymbol{\gamma}, (\boldsymbol{v} - \boldsymbol{v}_D) \right\rangle = 0 \\ b\left(\boldsymbol{v}^{n+1}, q\right) = 0 \end{cases} \tag{12}$$

Similarly, the matrix form is

$$\begin{bmatrix} \mathbf{M}/dt & \mathbf{A}^T & \mathbf{G}^T \\ \mathbf{G} & \mathbf{A} & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \mathbf{v} \\ \boldsymbol{\lambda} \\ \mathbf{p} \end{Bmatrix} = \begin{Bmatrix} \mathbf{M}\boldsymbol{v}_{int}^{n+1}/\triangle t \\ \mathbf{b}_2 \\ \mathbf{0} \end{Bmatrix} \tag{13}$$

where $\mathbf{G}$ is the same as we have seen in the steady Navier-Stokes problem.
This system of equations can be solved by LU decomposition of the left hand matrix. Therefore, we only need to solve

$$\mathbf{LU} \begin{Bmatrix} \mathbf{v} \\ \boldsymbol{\lambda} \\ \mathbf{p} \end{Bmatrix} = \begin{Bmatrix} \mathbf{M}\boldsymbol{v}_{int}^{n+1}/\triangle t \\ \mathbf{b}_2 \\ \mathbf{0} \end{Bmatrix} \tag{14}$$

3 Numerical results

Numerical studies are applied on the cavity flow with $R_e = 100$. The mesh and position of unkowns are shown in Figure 3. The velocity is defined in a 9 node quadrilateral and pressure 4 node quadrilateral.

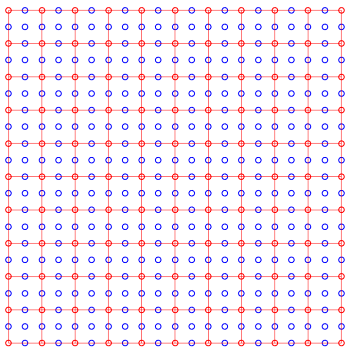
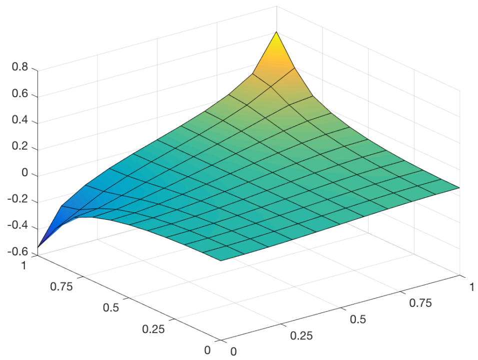


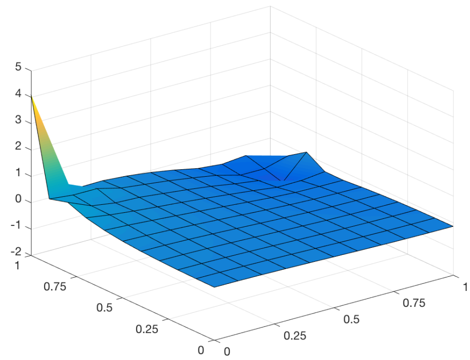
Figure 3: Uniform mesh

3.1 Explicit method

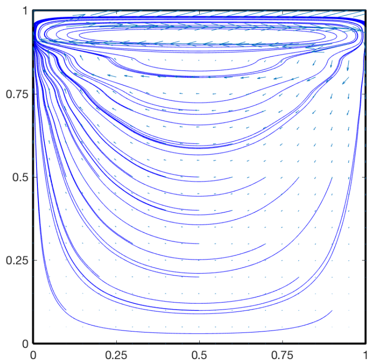
Figure 4 shows the pressure and streamline plots from the explicit method with time step equal to 0.1. It shows that for large time step, the explicit method is unstable and give unreasonable results.



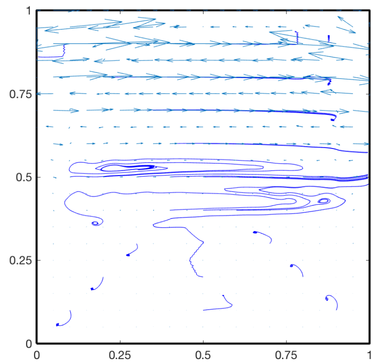
(a) pressure $t = 0.1$



(b) pressure $t = 0.4$



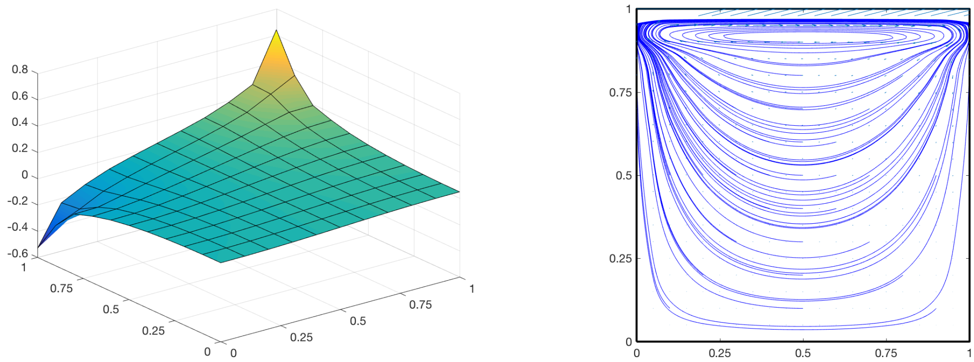
(c) streamlines $t = 0.1$



(d) streamlines $t = 0.4$

Figure 4: Plots of explicit method with $\triangle t = 0.1$

If we refine the time step $\triangle t = 0.01$, stable results will be achieved. Therefore, the explicit method is conditional stable, which requires a very fine time discretization.

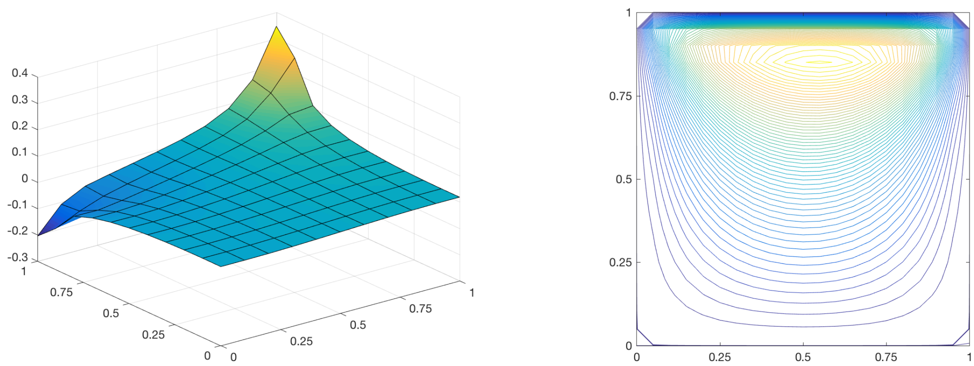


(a) pressure $t = 0.4$ (b) streamlines $t = 0.4$

Figure 5: Plots of explicit method with $\triangle t = 0.01$

3.2 Fully-implicit method

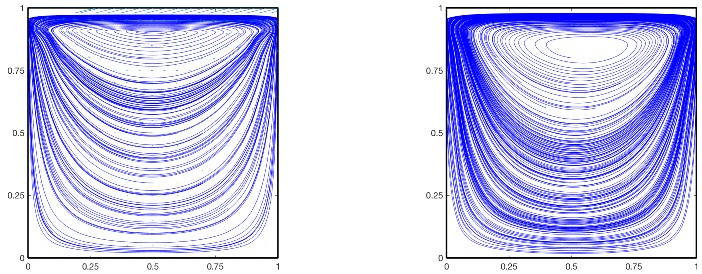
Figure 7 illustrates that for large time step $\triangle t = 0.1$, the results is accurate enough to depict the physical features of the cavity flow. In fact, the implicit methods are unconditionally stable. Therefore, it is preferable to apply semi-implicit and fully-implicit methods when the mesh is relatively coarse.



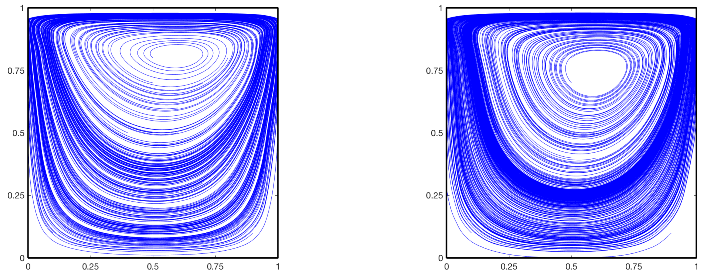
(a) pressure $t = 0.4$ (b) streamlines $t = 0.4$

Figure 6: Plots of fully-implicit method with $\triangle t = 0.1$

As we can see from Figure 7, the fluid in the domain gradually begin to rotate due to the imposed velocity at the top boundary as time passes. When the final time is very large, it tend to be the same as the results we obtained from the steady Navier-Stokes problem as shown in Figure 2.



(a) streamlines $t = 0.1$ (b) streamlines $t = 0.5$

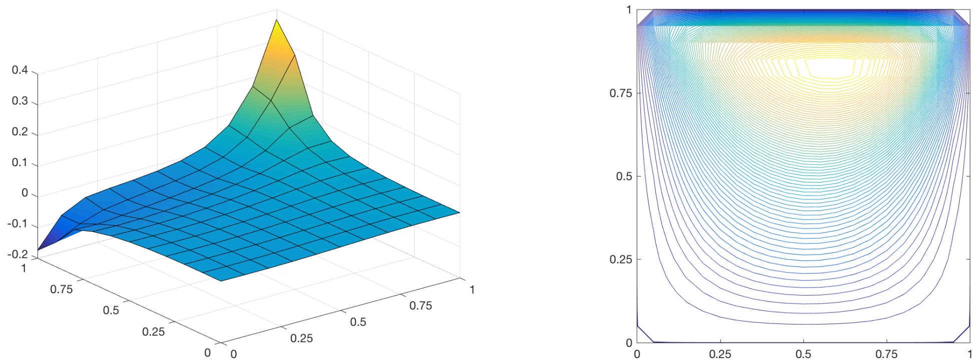


(c) streamlines $t = 1$ (d) streamlines $t = 5$

Figure 7: Evolution of streamlines with fully-implicit method $\triangle t = 0.1$

3.3 Semi-implicit method

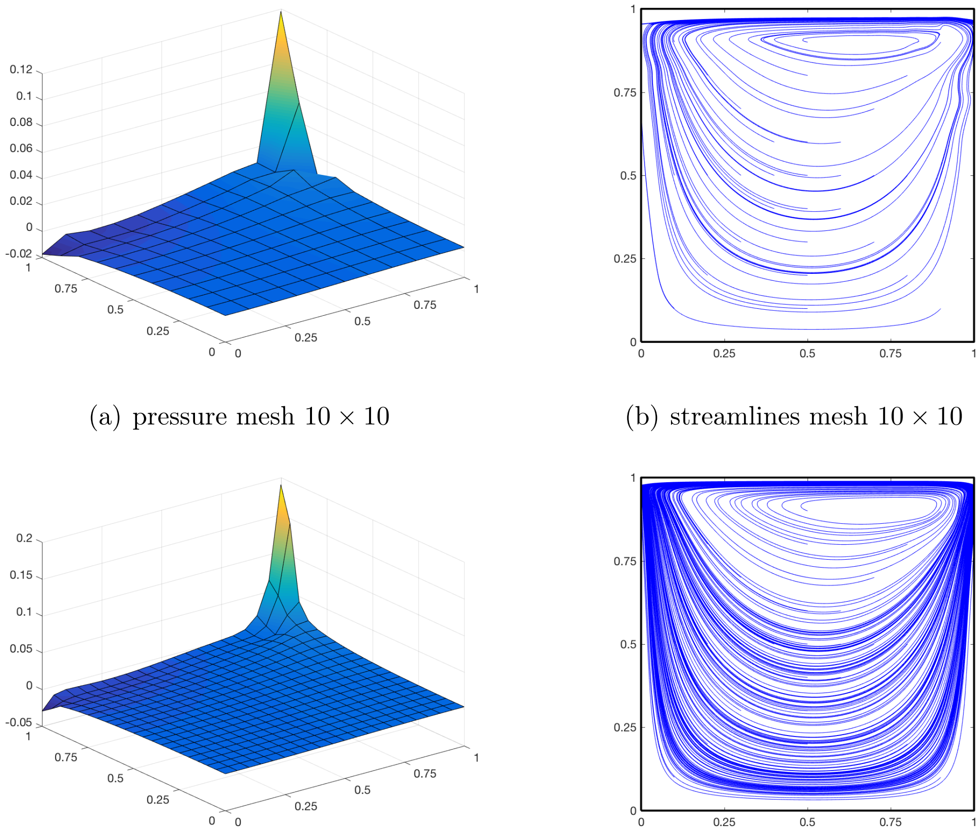
The semi-implicit method is also unconditionally stable. For example, as shown in Figure 8, the method gives reasonable results even with large time step $\triangle t = 0.1$.



(a) pressure $t = 0.5$ (b) streamlines $t = 0.5$

Figure 8: Plots of fully-implicit method with $\triangle t = 0.1$

If the Reynolds number is very high, for example $R_e = 1000$, the coarse mesh will not be able to capture the boundary layer. As we can see from Figure 9, great perturbation happens in the streamlines and the pressure field with mesh 10×10. With mesh 20×20, the results are more accurate as it simulate the transition in the boundary layer better. However, finer mesh is still required as the change of pressure in the corner elements are too agile.



(a) pressure mesh 10 × 10

(b) streamlines mesh 10 × 10

(c) pressure mesh 20 × 20

(d) streamlines mesh 20 × 20

Figure 9: Plots of semi-implicit method with $Re = 1000$