

FEF Assignment 4

Unsteady Navier Stokes

Hanna, John

1 Introduction

The Unsteady Navier Stokes equation describes a flow where both inertial and viscous forces play a role in the flow. The classical Galerkin method can be used to spatially discretize the problem, however, stabilization should be applied in some cases. Time can be discretized using the famous θ methods, or using Chorin-Temam projection method.

2 Problem description and summary of the methods

The problem to be solved is a square domain $1*1$ where 3 sides are fixed, while the upper one is not fixed having a prescribed velocity in the x-direction of value 1. This provides a discontinuity in the boundary conditions at the two upper corners which results in a singularity in the pressure solution as shown in the results section. The pressure is set to zero at the lower left corner. The initial velocity field is set to zero at time 0.

The differential equation that is been solved is given as:

$$\begin{aligned}v_t - \nu \nabla^2 v + (v \cdot \nabla)v + \nabla p &= b \quad \text{in } \Omega \\ \nabla \cdot v &= 0 \quad \text{in } \Omega \\ v_0 &= 0 \quad \text{in } \Omega \text{ and } t = 0 \\ v &= v_D \quad \text{in } \Gamma_D \text{ and } t \in]0, \text{inf}[\\ -pn + \nu(n \cdot \nabla)v &= t \quad \text{in } \Gamma_N \text{ and } t \in]0, \text{inf}[\end{aligned}$$

2.1 Semi-Implicit first order monolithic Scheme

The problem is spatially discretized using the Galerkin method leading to:

$$\begin{aligned}M\dot{u} + [K + C(u)]u + Gp &= f \\ G^T u &= h \end{aligned}$$

using the semi-implicit θ method, in which the convection matrix is calculated explicitly from the last step, we can obtain.

$$\begin{aligned}M \frac{u^{i+1} - u^i}{\Delta t} + [K + C(u)]u^{i+1} + Gp &= f \\ G^T u^{i+1} &= h \end{aligned}$$

rewriting the equation in matrix form to solve for the increments, we obtain

$$\begin{bmatrix} M + \Delta t \theta (K + C) & \Delta t \theta G \\ G^T & 0 \end{bmatrix} \begin{bmatrix} \Delta u \\ \Delta p \end{bmatrix} = \begin{bmatrix} \Delta t (F - [K + C]u^i - Gp^i) \\ 0 \end{bmatrix}$$

This system of equations is solved for each time increment.

2.2 Chorin-Temam projection method

This method depends on calculating an intermediate velocity step without regard to the pressure as follows after arrangement:

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

Then the velocity and pressure at the next time step is calculated as:

$$\begin{aligned} Mu^{i+1} + \Delta t G p^{i+1} &= Mu^* \\ G^T n^{i+1} &= 0 \end{aligned}$$

which can be rearranged in matrix form as

$$\begin{bmatrix} M & \Delta t G \\ G^T & 0 \end{bmatrix} \begin{bmatrix} u^{i+1} \\ p^{i+1} \end{bmatrix} = \begin{bmatrix} Mu^* \\ 0 \end{bmatrix}$$

3 Results and Discussion

The results were obtained using $dt = 0.002$ and Q2Q1 elements.

3.1 Semi-Implicit method

The value of θ was chosen to be 1.

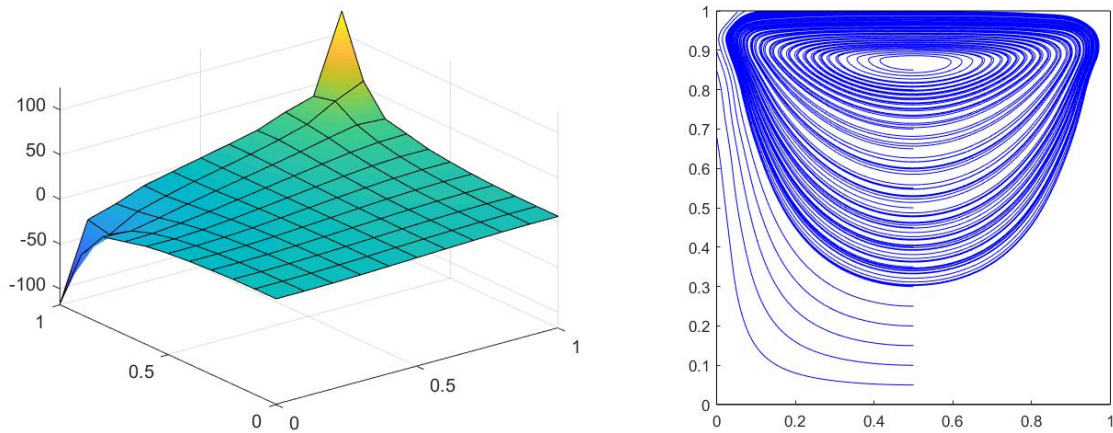


Figure 1: Pressure distribution and velocity contours at $t = 0.002$

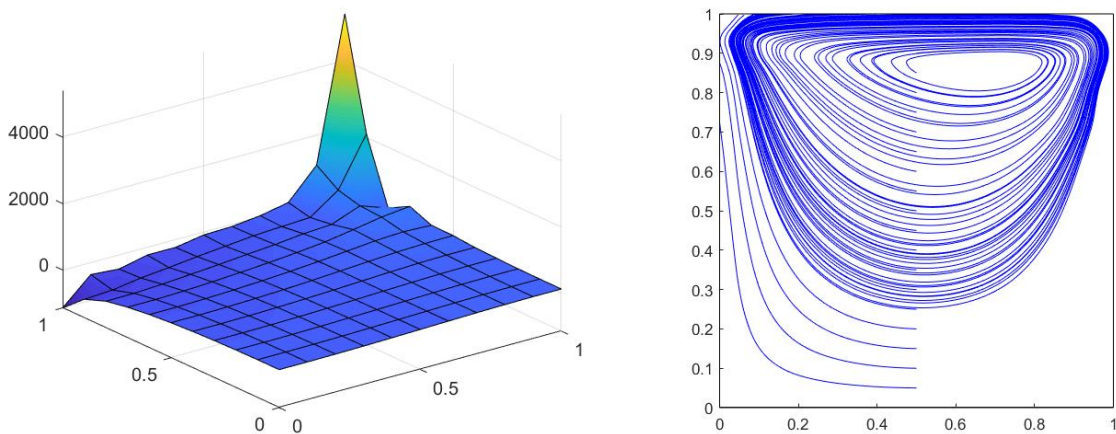


Figure 2: Pressure distribution and velocity contours at $t = 0.08$

As can be noted from the above figures, the pressure field is smooth at the beginning, then as time goes the discontinuities begin to appear with small perturbations near it. Better results can be obtained by using stabilization techniques.

3.2 Chorin-Temam projection method

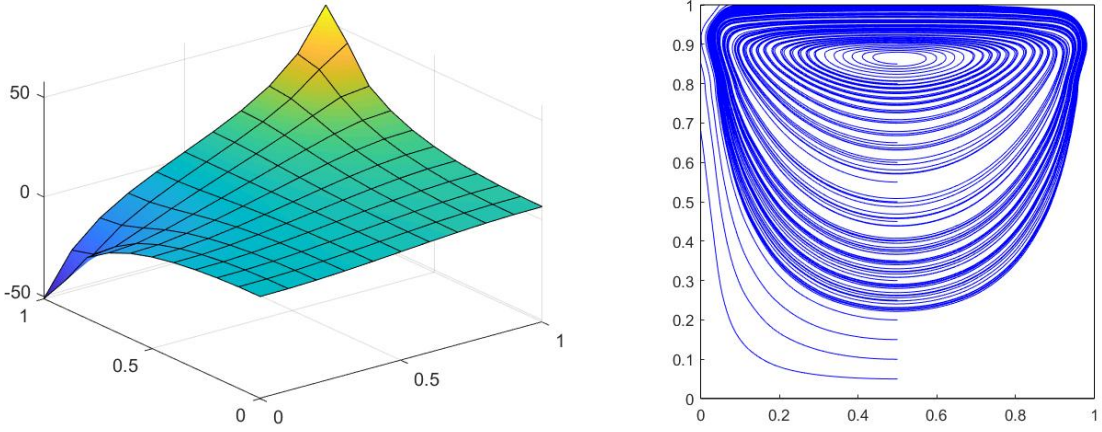


Figure 3: Pressure distribution and velocity contours at $t = 0.002$

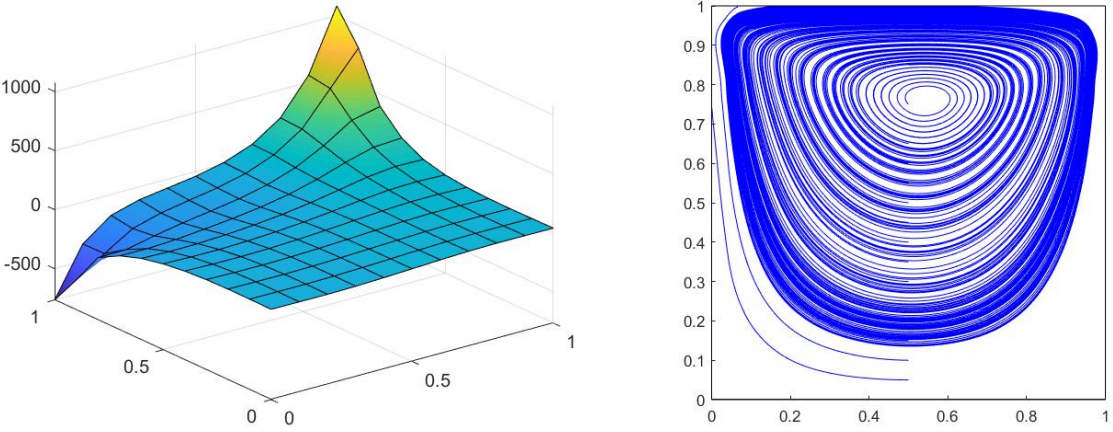


Figure 4: Pressure distribution and velocity contours at $t = 0.08$

Similar results are obtained for this method. However, the solution is more smooth and no stabilization is needed.

4 Appendix (MATLAB Codes)

4.1 Semi-Implicit Method

```
C = ConvectionMatrix(X,T,referenceElement ,velo );
Cred = C(dofUnk ,dofUnk );

fred = fred - (K(dofUnk ,dofDir)*valDir + C(dofUnk ,dofDir)*valDir );

A = [Mred+dt*(Kred+Cred) dt*Gred';
     Gred zeros(nunkP)];

B = [dt*(fred - (Kred+Cred)*veloVect(dofUnk)-Gred'*pres); zeros(nunkP,1)];

solInc = A\B;

veloInc = zeros(ndofV,1);
veloInc(dofUnk) = solInc(1:nunkV);
presInc = solInc(nunkV+1:end);
velo = velo + reshape(veloInc,2,[],)';
pres = pres + presInc;
```

4.2 Chorin-Temam projection method

```
C = ConvectionMatrix(X,T,referenceElement ,velo );
Cred = C(dofUnk ,dofUnk );

velored = veloVect(dofUnk );
fred = fred - (K(dofUnk ,dofDir)*valDir + C(dofUnk ,dofDir)*valDir );

A = Mred + dt*(Kred+Cred);
B = Mred*velored + dt*fred;

step1 = A\B; % Intermediate Velocity

Atot = [Mred dt*Gred';
        Gred zeros(nunkP)];
Btot = [Mred*step1; zeros(nunkP,1)];

sol = Atot\Btot; % Final solution

velosol = zeros(ndofV,1);
velosol(dofUnk) = sol(1:nunkV);
presol = sol(nunkV+1:end);
velo = reshape(velosol,2,[],)';
veloVect = reshape(velo',ndofV,1);
```