**CIMNE. International Centre for Numerical Methods in Engineering**

**Master of Science in Computational Mechanics**

**Universitat Politècnica de Catalunya**

# Finite elements in fluids
# Homework #5

**Submitted By**

**Aly Youssef**

## Problem description

This report is aimed at analyzing the methods used to solve an unsteady Navier-Stokes problem which describes general viscous unsteady Newtonian flows. A constraint to the Navier-Stokes problem is created through the assumption of incompressibility of the fluid. Solving such problem using finite elements provide some difficulties. This is due to the fact that pressure is acting as a Lagrange multiplier. Another difficulty in solving the Navier-Stokes problem is the nonlinear convective term. This term causes instability in the solution. In addition, the nonlinearity of the convective term requires iterative methods in order to solve the problem. In addition, time is a variable to be considered in the case of an unsteady problem. The Galerkin method is applied for the spatially discretize of the problem. Time could be discretized using several methods. In this report the θ-methods and the Chorin-Temam projection method are used.

The problem in hand is described as a one*one square domain. The boundary conditions (BC) are as follows: three sides are fixed, while the upper side has a prescribed velocity of magnitude one in the x-direction. The pressure is set to zero at the lower left corner. The described BCs induce a discontinuity at the upper two corners. This results in a singularity in the pressure solution.

The problem is governed by the following differential equation:

$$\begin{aligned}
&\mathbf{v}_t - \nu\nabla^2\mathbf{v} + (\mathbf{v}\cdot\nabla)\mathbf{v} + \nabla p = \mathbf{b} && \text{in } \Omega\\
&\nabla\cdot\mathbf{v} = 0 && \text{in } \Omega\\
&\mathbf{v} = \mathbf{v}_D && \text{on } \Gamma_D\\
&\mathbf{n}\cdot\sigma = \mathbf{t} && \text{on } \Gamma_N
\end{aligned}$$

*Equation 1 Governing differential equation*

The Galerkin special discretization leads to the following system of equations:

$$\begin{aligned}
&\mathbf{M}\dot{\mathbf{u}}(t) + [\mathbf{K} + \mathbf{C}(\mathbf{v})]\mathbf{u} + \mathbf{G}p = \mathbf{b}\\
&\mathbf{G}^T\mathbf{u}(t) = 0\\
&\mathbf{u}(\mathbf{0}) = \mathbf{v}(0) - \mathbf{v}_D(0)
\end{aligned}$$

*Equation 2 System of equations discretized in space to be solved*

## Semi-implicit first order monolitic scheme

The first method to be implemented is the semi-implicit discretization scheme. This method utilizes a procedure based on the θ-family of methods. For a single-step method, while neglecting the truncation errors, the time discretization scheme takes the following from:

$$\frac{\boldsymbol{v}^{n+1}-\boldsymbol{v}^{n}}{\Delta t}+\mathcal{C}(\boldsymbol{v})+\mathcal{K}(\boldsymbol{v}^{n+\theta})+\boldsymbol{\nabla} p^{n+\theta}=\boldsymbol{f}^{n+\theta}$$

$$\boldsymbol{\nabla}\cdot\boldsymbol{v}^{n+\theta}=0$$

*Equation 3 Time discretization*

It could be observed that the above equation varies depending on the parameter θ. Its value is defined in the interval [0; 1]. This determines whether the behavior of the scheme is fully explicit (θ = 0) or fully implicit (θ = 1). For a value of θ => 0.5 the method unconditionally stable; however, consistency is not assured. For a value of θ = 0.5 (Crank-Nicolson method) the solution is unconditionally stable, and it is second order accurate. The system in matrix notation takes the following form:

$$\begin{bmatrix}\boldsymbol{M}+\theta\Delta t(\boldsymbol{K}+\boldsymbol{C}^{n}) & \Delta t\theta\boldsymbol{G}\\ \boldsymbol{G}^{T} & 0\end{bmatrix}\begin{bmatrix}\Delta\boldsymbol{u}\\ \Delta\boldsymbol{p}\end{bmatrix}=\begin{bmatrix}\Delta t\big(\boldsymbol{b}-(\boldsymbol{K}+\boldsymbol{C}^{n})\boldsymbol{u}^{n}-\boldsymbol{G}p^{n}\big)\\ \boldsymbol{0}\end{bmatrix}$$

*Equation 4 System in matrix form*

## Choric-Temam projection method

The second method to be implemented is the Chorin-Temam method. This is a two-step method that calculates the velocity and pressure fields independently. The first step takes into account the viscous and convective terms only in order to calculate an intermediate velocity. This is shown in the following equation:

$$\mathbf{M}_{1}\left(\frac{\mathbf{v}_{\text{int}}^{n+1}-\mathbf{v}^{n}}{\Delta t}\right)+\big(\mathbf{C}(\boldsymbol{v}^{*})+\mathbf{K}\big)\mathbf{v}_{\text{int}}^{n+1}=\mathbf{f}^{n+1}$$

*Equation 5 Equation for the first step*

This intermediate velocity is used in the second step in order to obtain the final velocity and pressure fields, while taking into account incompressibility. This is shown in the following equation:

$$\mathbf{M}_2\left(\frac{\mathbf{v}^{n+1}-\mathbf{v}_{\text{int}}^{n+1}}{\Delta t}\right)-\mathbf{G}^T\mathbf{p}^{n+1}=\mathbf{0}$$

$$\mathbf{G}\mathbf{v}^{n+1}=\mathbf{0}$$

*Equation 6 Equation for the second step*

## Results and discussion

In this analysis, a comparison between both time discretization methods is due. The problem in hand is solved using a 10x10 mesh using Q2Q1 elements. This is done to ensure that no stabilization methods would be required as the LBB condition is met using this type of element. This comparison is to be conducted for different times at a constant time increment equal to 0.1. It is noted that for the semi-implicit first order monolitic scheme a value of θ < 0.5 results in an unstable solution. This could be seen in Figure 1.

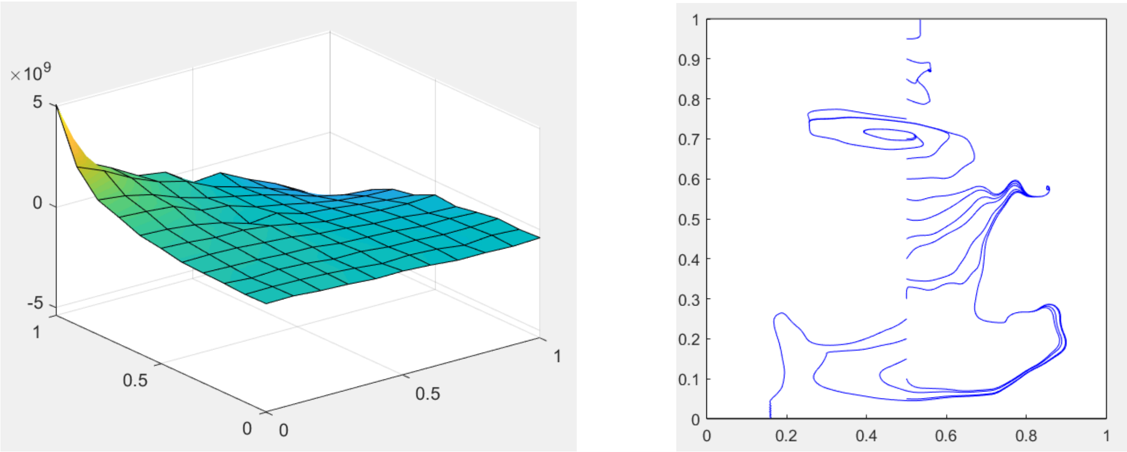

*Figure 1 Pressure distribution and velocity contours for ϑ = 0.1*

Thus, this comparison is conducted for θ = 0.5 (Crank-Nicolson) and θ = 1 (implicit). The total times are 0.5 (Figure 2), 1 (Figure 3) and 2 (Figure 4) seconds.

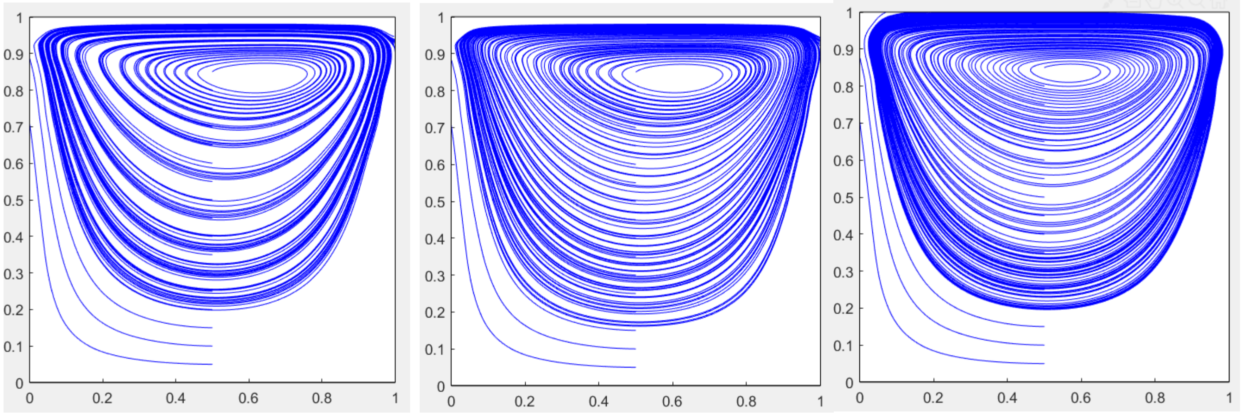

*Figure 2 Contour plot t = 0.5*

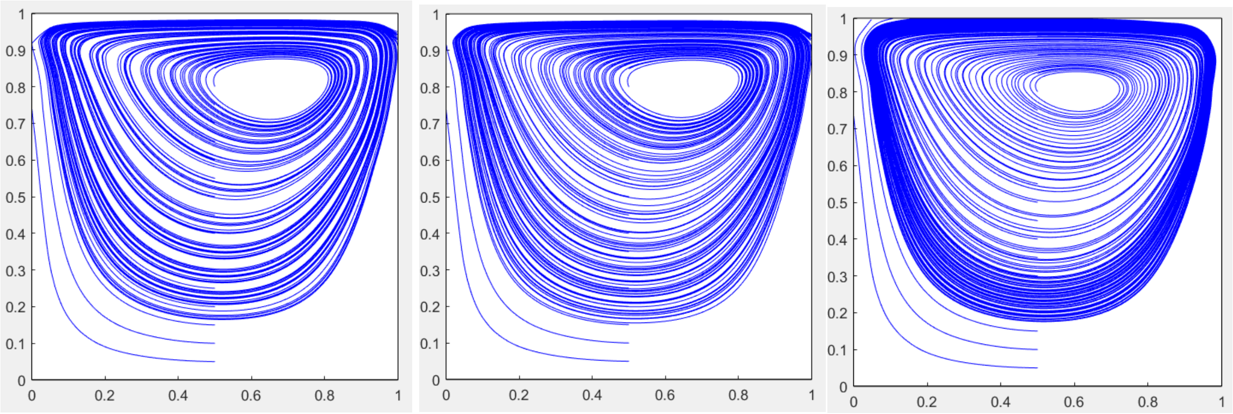

*Figure 3 Contour plot t = 1*

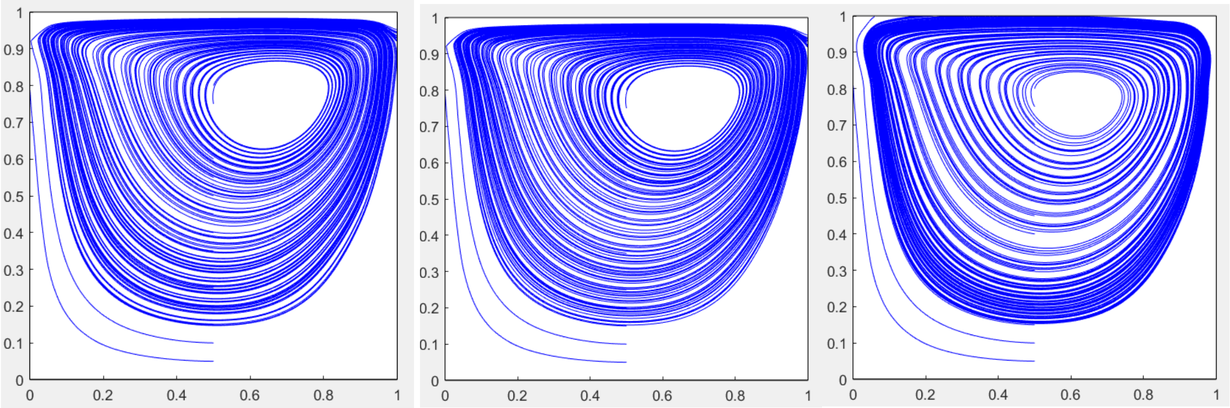

*Figure 4 Contour plot t = 2*

It could be seen from the above figures that the achieved results are almost the same regardless of the method used. The first difference that could be observed is the difference in the density of the streamlines; however, the difference is mnimal. The only significant difference between the two methods is the upper position of the flow. The Chorin-Temam show a slightly higher position compared to the semi-implicit solution.

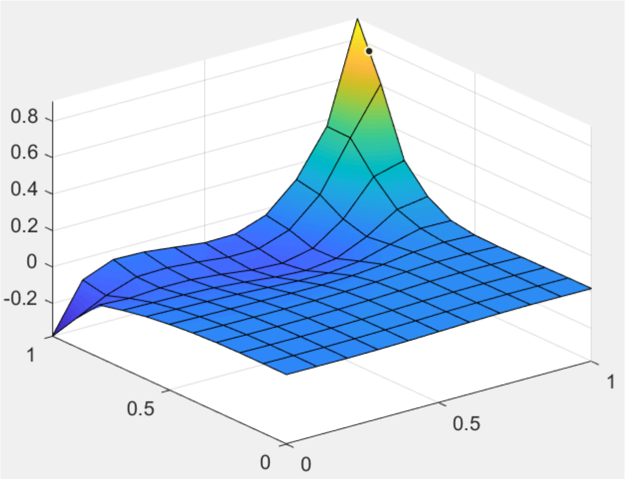

*Figure 5 Typical pressure distribution for t = 2*

All methods achieve a similar final pressure field. A typical one is obtained using the Chorin-Temam projection method at time equal to 2 (Figure 5).

# Code

## Semi-implicit first order monolitic scheme

### Initial conditions:

```
velo = zeros(ndofV/2,2);
y2 = dom(4);
nodesY2 = find(abs(X(:,2)-y2) < 1e-6);
velo(nodesY2,1) = 1;

save velocity.mat
```

### Modifications to the MainNavierStokes function:

```
% Semi-implicit first order monolitic scheme

if unstesdy_1 == 1

    fprintf('Semi-implicit first order monolitic scheme : \n')

    dt = delt_t;

    while iter < max_iter

        fprintf('Iteration = %d\n',iter);

        C = ConvectionMatrix (X,T, referenceElement , velo );
        Cred = C(dofUnk , dofUnk );
        fredn = fred - (K(dofUnk , dofDir )+C(dofUnk , dofDir ))* valDir ;

        % System of equations
        Atot = [ Mred+theta*dt*(Kred+Cred) dt*theta*Gred'
        Gred zeros(nunkP) ];
        btot = [dt *( fredn -( Kred + Cred )* veloVect ( dofUnk )-Gred' *
pres );
        zeros(nunkP,1) ];

        % Computation of velocity and pressure increment
        solInc = Atot \ btot ;

        % Update of the solution
        veloInc = zeros(ndofV ,1);
        veloInc ( dofUnk ) = solInc (1: nunkV );
        presInc = solInc ( nunkV +1: end );
        velo = velo + reshape (veloInc ,2 ,[])' ;
        pres = pres + presInc ;

        % Check convergence
        delta1 = max(abs(velo(:,1)));

        % Update variables for next iteration
        veloVect = reshape(velo',ndofV,1);
        sol0 = [veloVect(dofUnk); pres];
        iter = iter + 1;
        time = delt_t * iter;

        % Convergence plot
        plot(time,log(delta1),'r.','MarkerSize',20)
        xlabel('time')
        ylabel('Log(velocity)')
        hold on
    end
        hold off
end
```

## Choric-Temam projection method

Modifications to the iterative scheme in the mainNavierStokes:

```
% Choric-Temam projection method

if unstesdy_2 == 1

    fprintf('Choric-Temam projection method : \n')

dt = delt_t;

 while iter < max_iter

    fprintf('Iteration = %d\n',iter);

    C = ConvectionMatrix (X,T, referenceElement , velo );
    Cred = C(dofUnk , dofUnk );
    fredn = fred - (K(dofUnk , dofDir )+C(dofUnk , dofDir ))* valDir ;

    % FIRST STEP
    btot = dt* fredn + Mred * veloVect ( dofUnk );
    Atot = Mred +dt *( Cred + Kred );
    int = Atot \ btot ;

    % SECOND STEP
    btot = [ Mred *int; zeros(nunkP,1)];
    Atot = [ Mred Gred'*dt; Gred zeros(nunkP)];
    solInc = Atot \ btot ;

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

   % Check convergence
   delta1 = max(abs(velo(:,1)));

   % Update variables for next iteration
```

```
  veloVect = reshape(velo',ndofV,1);
  sol0 = [veloVect(dofUnk); pres];
  iter = iter + 1;

  % Convergence plot
  plot(iter,log(delta1),'r.','MarkerSize',20)
  xlabel('time')
  ylabel('Log(velocity)')
  hold on
end
  hold off
```