FINITE ELEMENT IN FLUIDS

Homework 5

Unsteady Navier-Stokes problem

Author: Cristina García Albela

MsC in Computational Mechanics

Unsteady Navier-Stokes problem

Navier – Stokes equation govern steady or unsteady, viscous incompressible flows. Here the unsteady case is going to be studied, whose governing equation and initial and boundary conditions are defined as

$$\begin{aligned}
&v_t - \nu\nabla^2 v + (v\cdot\nabla)v + \nabla p = b && in\ \Omega\times\,]0,T[\\
&\nabla\cdot v = 0 && in\ \Omega\times\,]0,T[\\
&v = v_D && on\ \Gamma_D\times\,]0,T[\\
&v(x,0) = v_0(x) && in\ \Omega
\end{aligned}$$

Spatial discretization

As usual, applying WRM and after some algebra the weak formulation is obtained, representing $\boldsymbol{w}$ and q the weighting functions for momentum equation and incompressibility condition respectably.

$$\int_\Omega \boldsymbol{w}\cdot\boldsymbol{v_t}d\Omega + \int_\Omega (\nabla\boldsymbol{w}):(\nu\nabla\boldsymbol{v})d\Omega + \int_\Omega \boldsymbol{w}\cdot(\boldsymbol{a}\cdot\nabla)\boldsymbol{v}d\Omega - \int_\Omega (\nabla\cdot\boldsymbol{w})pd\Omega = \int_\Omega \boldsymbol{w}\cdot\boldsymbol{f}d\Omega$$

$$\int_\Omega q\nabla\cdot\boldsymbol{v}d\Omega = 0$$

Introducing the approximation forms for velocity and pressures, as well as applying Galerkin formulation for their weighted functions $(\boldsymbol{w},q)$, the finite element discretization yields the following system of semi-discrete equations

$$\begin{cases}\boldsymbol{Mv_t} + [\boldsymbol{K} + \boldsymbol{C}(v(t))]\boldsymbol{v}(t) + \boldsymbol{Gp}(t) = \boldsymbol{f}(t, v(t))\\ \boldsymbol{G^T v}(t) = \boldsymbol{h}(t)\\ \boldsymbol{v}(0) = \boldsymbol{v_0} - \boldsymbol{v_D}(0)\end{cases}$$

where a new element appears with respect to the steady case, the standard mass matrix $\boldsymbol{M}$.

Time discretization

Working with a time dependent problem, spatial discretization is not enough. Between all the possibilities, two different ways to deal with time integration will be implemented: Monolithic schemes (Theta family methods) and Fractional-Steps method.

1. Theta family methods – First order approximation

Theta methods are suitable to solve the previous system of discrete equations. Having the initial definition of theta methods (without error term) together with the time dependent momentum equation, a new system of equations is achieved.

$$Theta\ methods \rightarrow\ \frac{v_i^{n+1} - v_i^n}{\Delta t} = \theta v_t^{n+1} + (1-\theta)v_t^n = \theta(v_t^{n+1} - v_t^n) - v_t^n$$

$$Time\ dependent\ momentum\ equation \rightarrow v_t = \frac{f - [K + C(v)]v - Gp}{M}$$

Substituting v_t in theta methods equation,

$$\frac{v_i^{n+1}-v_i^n}{\Delta t}=\theta\left[\frac{f-[K+C(v)]v^{n+1}-Gp^{n+1}}{M}-\frac{f-[K+C(v)]v^n-Gp^n}{M}\right]+\frac{f-[K+C(v)]v^n-Gp^n}{M}$$

$$\frac{\Delta \boldsymbol{v}}{\Delta \boldsymbol{t}}\boldsymbol{M}+\boldsymbol{\theta}\big[\big(\boldsymbol{K}+\boldsymbol{C}(\boldsymbol{v})\big)\Delta\boldsymbol{v}+\boldsymbol{G}\Delta\boldsymbol{p}\big]=\boldsymbol{f}-[\boldsymbol{K}+\boldsymbol{C}(\boldsymbol{v})]\boldsymbol{v}^{\boldsymbol{n}}-\boldsymbol{G}\boldsymbol{p}^{\boldsymbol{n}}$$

This new equation together with the incompressibility condition can be written as a matrix system, which will be implemented and solved using Matlab software.

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

Remark the fact that if the chosen pair of elements for velocity and pressure field do not match the LBB condition, stabilization procedures have to be used. It can be said that the biggest different from steady case is that no Picard or N-R methods are needed, making it easier to compute.

Here the first order ($\theta = 1$) semi-implicit ($\boldsymbol{C}(\boldsymbol{v}) = (\boldsymbol{v}^{\boldsymbol{n}}\cdot\nabla)\boldsymbol{v}^{\boldsymbol{n+1}}$) approximation is going to be implemented. Having in mind the steady case, the main aspects of general code are going to be shown, highlighting the introduction of time parameters (Figure 1) and the new definition of the final matrix system (Figure 4), from which velocity and pressure increments are obtained, updating the solution at each time step. The well-known equations for $\boldsymbol{K}, \boldsymbol{G}$ and $\boldsymbol{C}$ are implemented together with the new component, the mass matrix $\boldsymbol{M}$ (Figure 2 and Figure 3).

```
% Time discretization
tEnd = cinput('End time', 1.5);
nStep = cinput('Number of time-steps', 240);
dt = tEnd / nStep;
```

Figure 1. Time discretization parameters

```
  Me = Me + Ngp'*Ngp*dvolu;
  Ke = Ke + (Nx'*Nx+Ny'*Ny)*dvolu;
  Ge = Ge - NP_ig'*dN*dvolu;
  x_ig = N_ig(1:ngeom)*Xe;
  f_igaus = SourceTerm(x_ig);
  fe = fe + Ngp'*f_igaus*dvolu;
```

Figure 2. Element matrices and vector definitions

```
Ce = Ce + Ngp'*(v_ig(1)*Nx+v_ig(2)*Ny)*dvolu;
```

Figure 3. Convective matrix

```
%R.H.S as in steady NS case (velocity field)
fred_1 = fred - (K(dofUnk,dofDir)*valDir + C(dofUnk,dofDir)*valDir);
%A Matrix
Atot = [Mred+teta*dt*(Kred+Cred)   teta*dt*Gred'
    Gred   zeros(nunkP)];
%R.H.S due to time discretization
   B = [dt*(fred_1 - (Kred+Cred)*veloVect(dofUnk)-Gred'*pres);zeros(nunkP,1)];
% Computation of velocity and pressure increment
   solInc = Atot\B;
% Update the solution
   veloInc = zeros(ndofV,1);
   veloInc(dofUnk) = solInc(1:nunkV);
   presInc = solInc(nunkV+1:end);
   velo = velo + reshape(veloInc,2,[])';
   pres = pres + presInc;
```

Figure 4. Matrix system and solutions update

Stabilization - LBB condition

Working with velocity – pressure pairs which doesn't satisfy LBB condition a stabilization technique has to be implemented. As in Stokes problem GLS formulation will be used.

$$\int_{\Omega} \begin{bmatrix} \boldsymbol{w} \\ q \end{bmatrix} \cdot (\boldsymbol{\mathcal{L}}(\boldsymbol{v},p) - \boldsymbol{F}) d\Omega + \sum_{e} \int_{\Omega_e} \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} \boldsymbol{\mathcal{L}}(\boldsymbol{w},q) \cdot (\boldsymbol{\mathcal{L}}(\boldsymbol{v},p) - \boldsymbol{F}) d\Omega = \boldsymbol{0}$$

$$\tau_1 = \alpha_0 \frac{h^2}{4\nu} \qquad \tau_2 = 0 \qquad (\alpha_0 = \frac{1}{3}\, for\ linear\ elements)$$

Working with linear elements the second order terms goes to zero, so that the GLS does not affect the momentum equation. Together with τ_2 value, the new formulation reduces to

$$\begin{bmatrix} \boldsymbol{M} + \theta\Delta t\big(\boldsymbol{K} + \boldsymbol{C}(\boldsymbol{v})\big) & \Delta t \boldsymbol{G}^{\boldsymbol{T}} \Delta p \\ \boldsymbol{G} & \boldsymbol{L} \end{bmatrix} \begin{bmatrix} \Delta \boldsymbol{u} \\ \Delta p \end{bmatrix} = \begin{bmatrix} \Delta t[\boldsymbol{f} - [\boldsymbol{K} + \boldsymbol{C}(\boldsymbol{v})]\boldsymbol{v}^{\boldsymbol{n}} - \boldsymbol{G}p^{n}] \\ \boldsymbol{f}_{\boldsymbol{q}} \end{bmatrix}$$

where the new terms $\boldsymbol{L}$ and $\boldsymbol{f}_{\boldsymbol{q}}$ have to added to the previous code (Figure 5 and Figure 6).

```
Me = Me + Ngp'*Ngp*dvolu;
Ke = Ke + (Nx'*Nx+Ny'*Ny)*dvolu;
Ge = Ge - NP_ig'*dN*dvolu;
Le = Le + (nx'*nx + ny'*ny)*dvolu;
x_ig = N_ig(1:ngeom)*Xe;
f_igaus = SourceTerm(x_ig);
fe = fe + Ngp'*f_igaus*dvolu;
fqe = fqe – tau1*[nx;ny]'*f_igaus*dvolu;
```

Figure 5. Element matrix components – GLS

```
%R.H.S as in steady NS case (velocity field)
fred_1 = fred - (K(dofUnk,dofDir)*valDir + C(dofUnk,dofDir)*valDir);
%A Matrix
Atot = [Mred+teta*dt*(Kred+Cred)  teta*dt*Gred'
    Gred   L]
%R.H.S due to time discretization
   B = [dt*(fred_1 - (Kred+Cred)*veloVect(dofUnk)-Gred'*pres);fq];
% Computation of velocity and pressure increment
   solInc = Atot\B;
```

Figure 6. Matrix system - GLS

Results and conclusions

First order semi-implicit case is going to be solve for Q2Q1 and Q1Q1 velocity – pressure elements. Using 10 and 20 elements per side and a time step $dt = 0.015s$, results are going to be shown and compared.

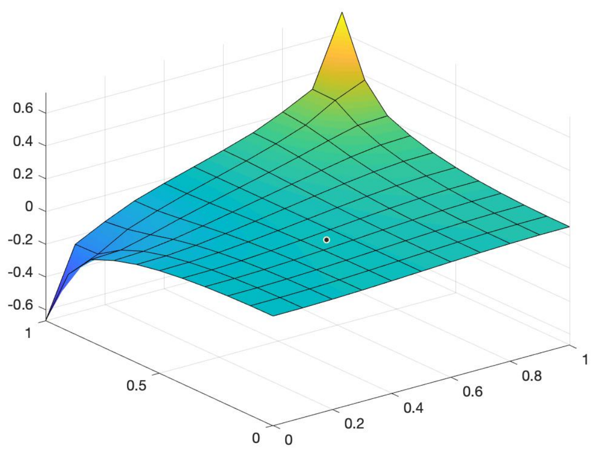
Figure 7. Pressure field – Q2Q1 10 elements, Initial t step

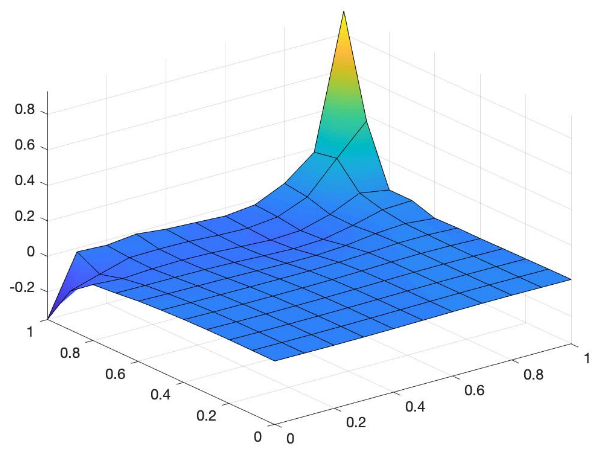
Figure 8. Pressure field – Q2Q1 10 elements, final t step

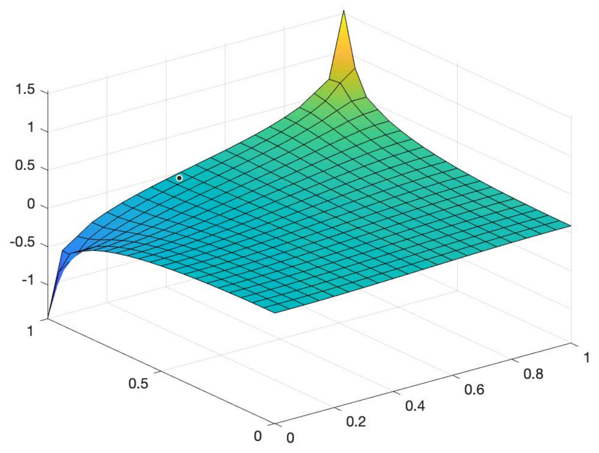
Figure 9. Pressure field – Q2Q1 20 elements, Initial t step

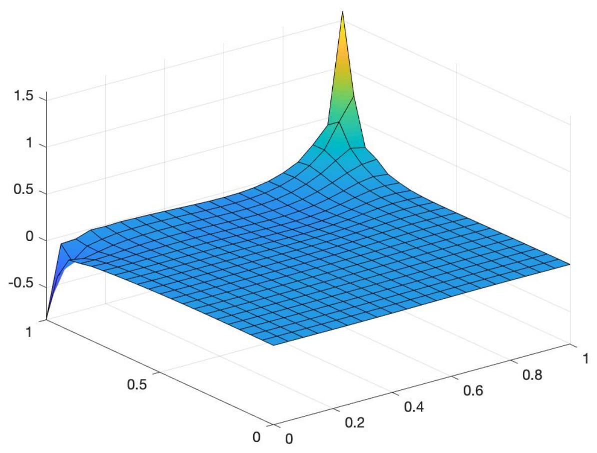
Figure 10. Pressure field – Q2Q1 20 elements , final t step

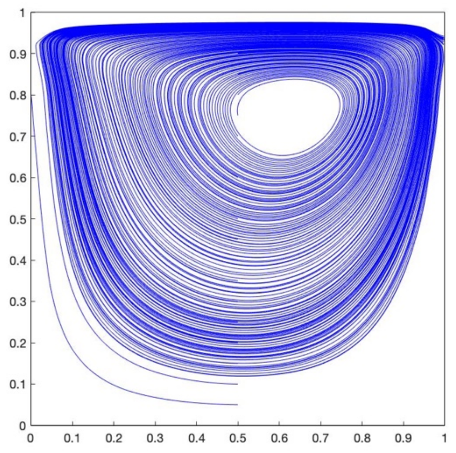
Figure 11. Streamlines – Q2Q1 10 elements

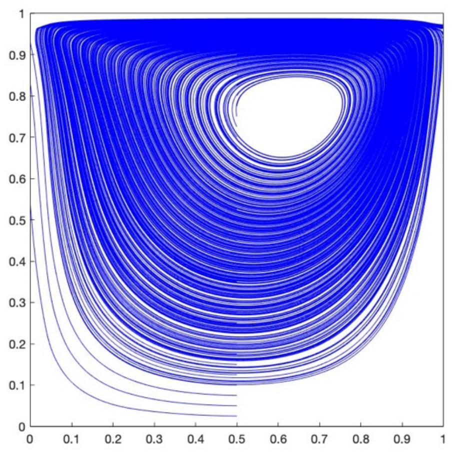
Figure 12. Streamlines – Q2Q1 20 elements

Figures above show the results for Q2Q1 elements, for which no stabilization technique is needed as they fulfil LBB condition. Left pictures represent the pressure field at first time iteration, whereas right hand side ones represent it achieved the total time. See how the boundary condition that contains the discontinuity changes as time increases. A refined mesh of 20 element, that implies more time and computational cost, is computed in order to improve the results that with 10 elements show small variations close to the boundary sides. The obtained final results remind to the steady case, suggesting that the implementation of the method has been successful.

Now, results for Q1Q1 elements are shown. Due to GLS stabilization term quite good results are achieved, in which the difference between a gross (Figure 13) and a refined (Figure 14) mesh is much more significant. This fact happened in Stokes problem too, where a 30x30 elements mesh was implemented in order to improve bad pressure field results in those zones close to the boundaries. For this type of elements combinations is worth to increase the computational cost in order to be sure about the validity of the results.

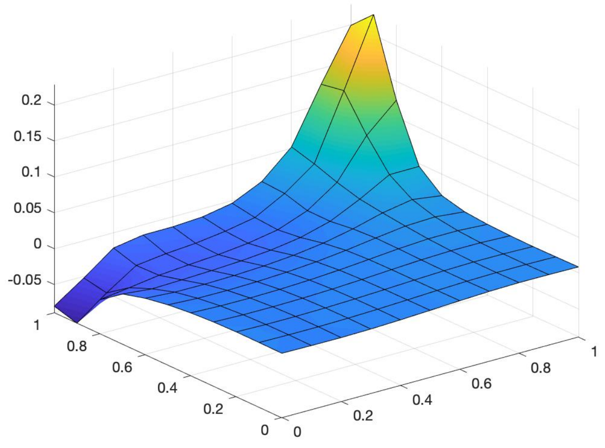

Figure 13. Pressure field – Q1Q1 10 elements GLS stabilization

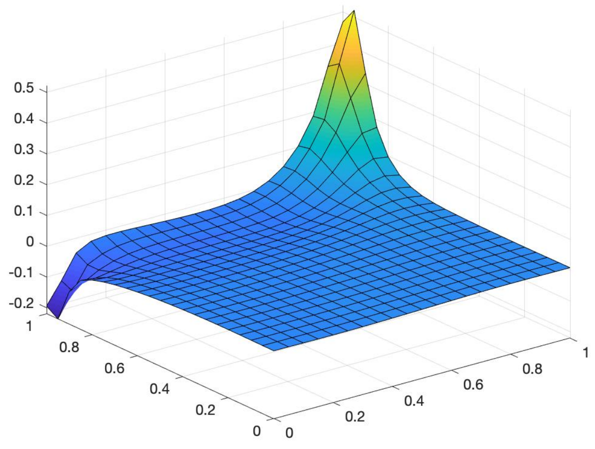

Figure 14. Pressure field – Q1Q1 20 elements GLS stabilization

2. Fractional step methods – Chorin-Temam projection method

Other typical methods for unsteady Navier-Stokes time discretization are the fractional-step methods, in which time is decomposed into two or more steps, dividing the initial problem into relatively easier substeps. Working with a two steps method, the first step that treats the convective and viscous terms can be implemented by an explicit algorithm, while the second step that works with pressure and incompressibility condition must be treated using an implicit time integration scheme. Here, a semi-implicit Chorin – Temam method is going to be studied to solve unsteady Navier-Stokes equations.

Chorin – Temam projection method

The projection method is based on compute velocity and pressure fields separately through the computation of an intermediate velocity. It includes two steps:

Step 1. First, viscous and convective terms in the N-S equations are treated. From the previous time step velocity field (v^n), an intermediate velocity is finding (v_{int}^{n+1}).

$$\begin{cases} \dfrac{v_{int}^{n+1} - v^n}{\Delta t} + (v^* \cdot \nabla)v^{**} - \nu\nabla^2 v^{**} = b^{n+1} & in\ \Omega \\ v_{int}^{n+1} = v_D^{n+1} & on\ \Gamma \end{cases}$$

where for semi-implicit method $v^* = v^n$ and $v^{**} = v^{n+1}$.
As for all the previous methods the weak form is obtained to construct a finite element version, that results in the following algebraic system which will be implement in a Matlab code. Note that $\boldsymbol{M}, \boldsymbol{K}$ and $\boldsymbol{C}$ will be defined as in theta methods case (Figure2 and Figure 3), appearing the principal difference in how the r.h.s. is defined (Figure 15).

$$\boldsymbol{M_1}\left(\frac{\boldsymbol{v}_{int}^{n+1} - \boldsymbol{v}^n}{\Delta t}\right) + \left(\boldsymbol{K} + \boldsymbol{C}(\boldsymbol{v^n})\right)\boldsymbol{v_{int}^{n+1}} = \boldsymbol{f^{n+1}}$$

```
%R.H.S as in steady NS case
fred_1 = fred - (K(dofUnk,dofDir)*valDir + C(dofUnk,dofDir)*valDir);

% STEP 1
B = dt*fred_1 + Mred*veloVect(dofUnk);
Atot = Mred+dt*(Cred+Kred);
int = Atot\B;
```

Figure 15. Step 1 implementation – C-T projection method

Step 2. Then, the final step velocity and the pressure are obtained solving

$$\begin{cases} \dfrac{\boldsymbol{v}^{n+1} - \boldsymbol{v}_{int}^{n+1}}{\Delta t} + \nabla p^{n+1} = 0 & in\ \Omega \\ \nabla \cdot v^{n+1} = 0 & in\ \Omega \\ n \cdot \boldsymbol{v}^{n+1} = n \cdot v_D^{n+1} & on\ \Gamma \end{cases}$$

The weak form is introduced, obtaining the finite element version and inducing the following system of equations. Highlight the fact that, this last system is quite similar to the Stokes problem case, changing the $\boldsymbol{K}$ matrix for the $\boldsymbol{M}$ matrix. Using once again the previous definitions for the different matrices (Figure 2), the 2nd step of C-T projection method and the final updating are implemented (Figure 16).

$$\begin{bmatrix} M_2 & \Delta t G^T \\ G & 0 \end{bmatrix} \begin{bmatrix} v^{n+1} \\ p^{n+1} \end{bmatrix} = \begin{bmatrix} M_2 v_{int}^{n+1} \\ 0 \end{bmatrix}$$

```
% STEP 2
B_2 = [Mred*int; f_q];
Atot_2 = [Mred Gred'*dt; Gred L];
solInc = Atot_2\B_2;

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

Figure 16. Step 2 and updating implementation – C-T projection method

Stabilization - LBB condition

Once again, working with element-pairs that do not fulfil the LBB condition, means to introduce a stabilization term, GLS formulation in this case. Following the same and having in mind that it will be implemented to work just with linear elements (second order terms vanish), the following scheme is obtained

$$\begin{bmatrix} M_2 & \Delta t G^T \\ G & L \end{bmatrix} \begin{bmatrix} \boldsymbol{v}^{n+1} \\ \boldsymbol{p}^{n+1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{M_2 v}_{int}^{n+1} \\ \boldsymbol{f_q} \end{bmatrix}$$

Applying the previous $\mathbf{L}$ and $\boldsymbol{f_q}$ definitions (Figure 5), stable results will be obtained for Q1Q1 elements discretization.

Results and conclusions

Semi-implicit Chorin-Temam case is going to be solved for Q2Q1 and Q1Q1 velocity – pressure elements, using 10 and 20 elements per side and a time step $dt = 0.015s$. Pressure and velocity results are almost the same than for semi-implicit theta method, so that just final time step results are going to be shown.

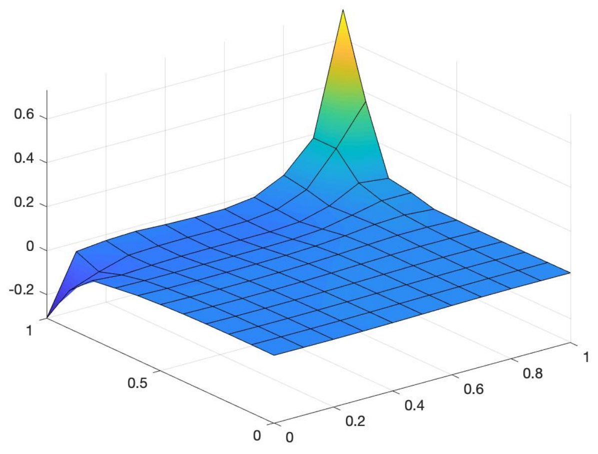

Figure 17. Pressure field – Q2Q1 10 elements, Final t step

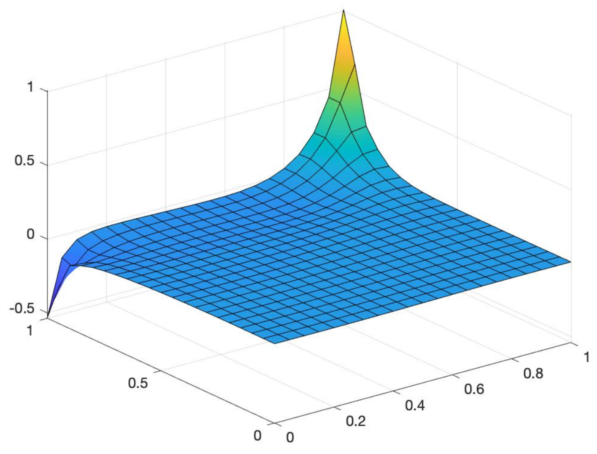

Figure 18. Pressure field – Q2Q1 20 elements, final t step

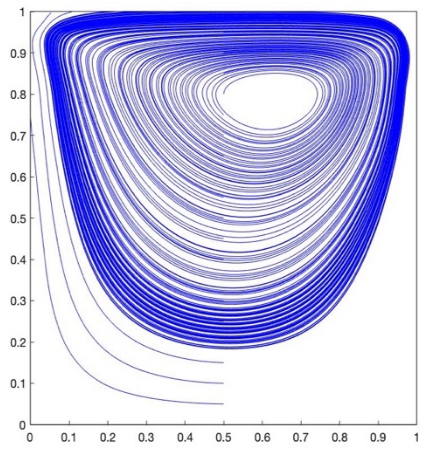

Figure 19. Streamlines – Q2Q1 10 elements

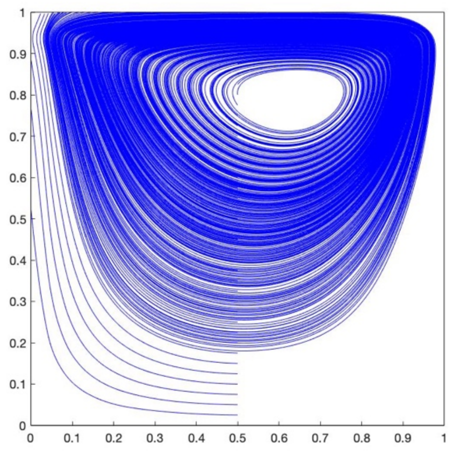

Figure 20. Steamlines – Q2Q1 20 elements

For Q2Q1 discretization once again small perturbations appear close to the boundaries, that almost disappear when the number of elements increase. Depending on the case, it should be assessed whether higher precision, which increases computational cost, is worth it.

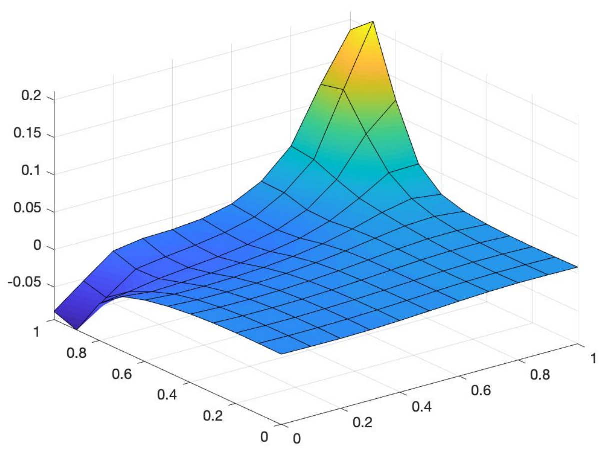

Figure 21. Pressure field – Q1Q1 10 elements GGLS

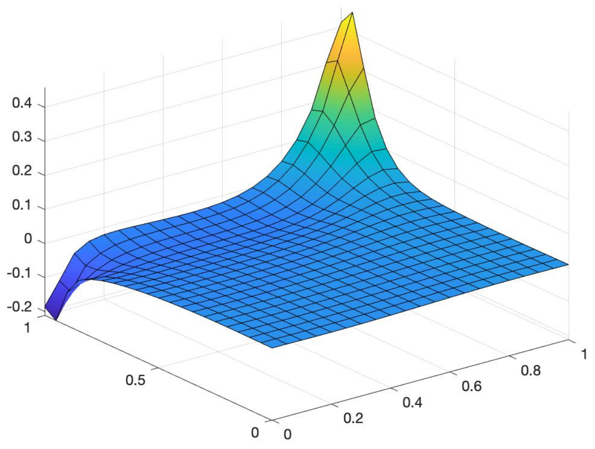

Figure 22. Pressure field – Q1Q1 10 elements GLS

Finally, results for Q1Q1 discretization. The difference in final results accuracy in those zones close to the boundaries, highlining the boundary which includes the discontinuity, compensates the extra computational cost.