Finite Elements in Fluid

Homework 5a: Stokes numerical examples

Ye Mao

Ye Mao, mao.ye@estudiant.upc.edu
Master of Numerical methods on engineering - Universitat Politècnica de Catalunya

1. INTRODUCTION

   Stokes equations

$$-\nu\,\nabla^2 v + \nabla p = b \qquad in\Omega$$

$$\nabla\bullet v = 0 \qquad in\Omega$$

   Weak form:

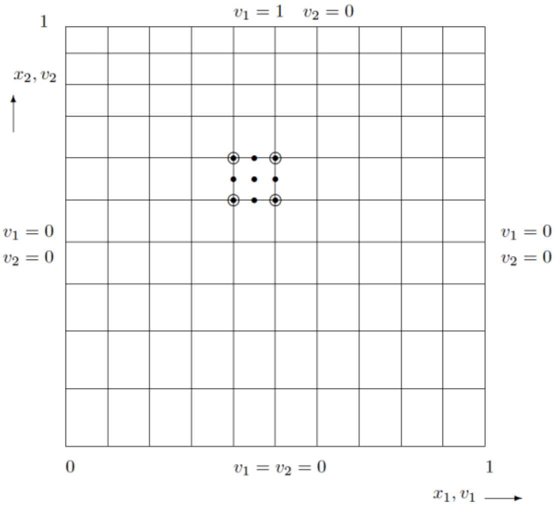
$$\begin{cases}\int_\Omega \nabla w : \nu\,\nabla v d\Omega - \int_\Omega p\,\nabla\bullet w d\Omega - = \int_\Omega p\,\nabla\bullet w d\Omega - w\bullet b d\Omega & \forall w\in \boldsymbol{v}\\ \int_\Omega q\,\nabla\bullet v d\Omega = 0 & \forall q\in \boldsymbol{Q}\end{cases}$$

   Galerkin discretization

$$\begin{pmatrix} K & G\\ G^T & 0\end{pmatrix}\begin{pmatrix} u\\ p\end{pmatrix} = \begin{pmatrix} f\\ h\end{pmatrix}$$

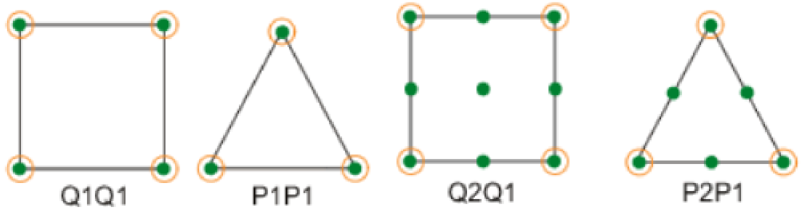
2. OBJECTIVE
   2.1 Describe physically the problem based on the value of BC.
   2.2 Solve the problem using different element types Q1Q1, Q2Q1, P1P1, P2P1. Find the stable ones.
   2.3 Program a stabilization method on the unstable elements.

3. METHODOLOGY AND RESULTS
   3.1 Physical Description of the problem.
   This example based on the value of BC can be considered as a classical benchmark test for incompressible flow: the cavity flow problem,



It can be seen obviously that it poses a close solution with velocity field $v(v_1, v_2)$ and pressure filed p with boundary conditions in the square domain $\Omega = [0,1]\mathrm{x}[0,1]$. The provided BC is the divergence free velocity distribution. There exists a discontinuity in the boundary conditions at the upper corners of the cavity. Dirichlet boundary conditions are imposed on every boundary. Pressure is known at an arbitrary point in the domain. In this case, it is set in the lower left corner of the cavity where P equal to 0.

The problem is discretized with $10 \mathrm{~x~} 10$ nodes in square, 10 elements in each direction. We choose four different types of elements to analysis this problem, which are Q1Q1, P1P1, Q2Q1 and P2P1.



The key point for these elements is to show stability while reach the compatibility condition as LBB or inf-sup condition to avoid the oscillations of unphysical node-to-node pressure. According to this condition, pressure approximation must be at least one order lower than velocity approximation in each element. The use of GLS not only just enable non-oscillatory solution of high Reynold number problems with reasonably fine meshes, but also enable the use of equal order interpolation for velocity and pressure. They avoid the zero diagonal entries of the global system of equations as aforementioned. GLS is a residual based on stabilization. If the original FEM solution is good enough, it is the free based on unphysical oscillation. Then, the useful affection by this method is negligible.

To overcome the problem which having 0 entries on the main diagonal of the algebraic system of equations, we could apply Galerkin Least-Squares methond.

3.2 Compare the results.

3.2.1 Q1Q1 Element,

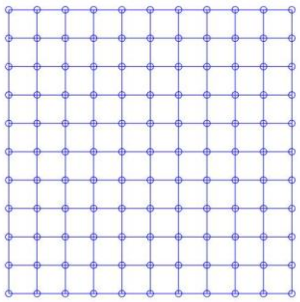


Figure 1. Q1Q1 Element scheme

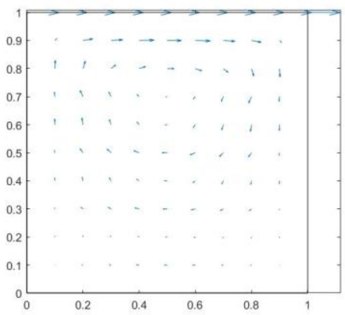
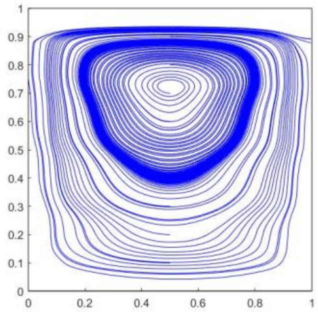


Figure 2. Q1Q1 Velocity Vectors



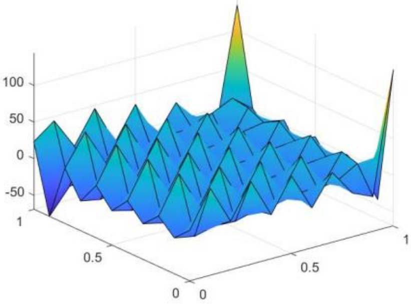
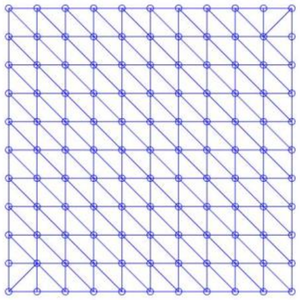


Figure 3. Q1Q1 Streamlines

Figure 4. Q1Q1 Pressure Field

The continuous bilinear velocity and continuous pressure element does not satisfy LBB condition. As per expectation, it shows inaccuracy in term of pressure. The solution oscillates from element to element across all the field.

3.2.2 P1P1 Element,



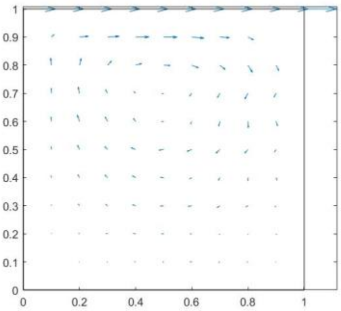
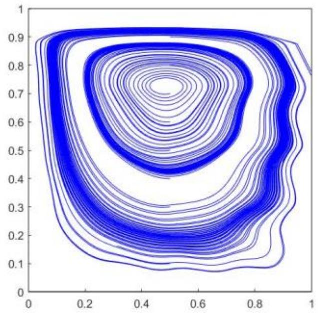


Figure 5. P1P1 Element scheme

Figure 6. P1P1 Velocity Vectors



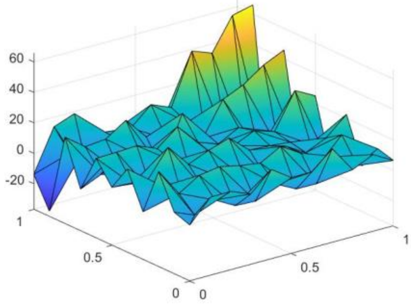


Figure 7. P1P1 Streamlines

Figure 8. P1P1 Pressure Field

The continuous triangular velocity and triangular pressure element does not satisfy LBB condition. The result is highly inaccurate in term of pressure field. Specially, it has larger element to element oscillation at corners. The streamlines are discontinuity in the same time.

3.2.3 Q2Q1 Element and P2P1 Element

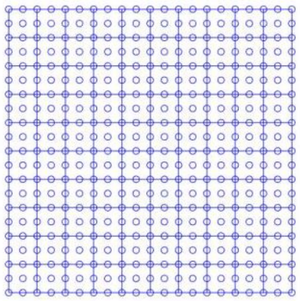


Figure 9. Q2Q1 Element scheme

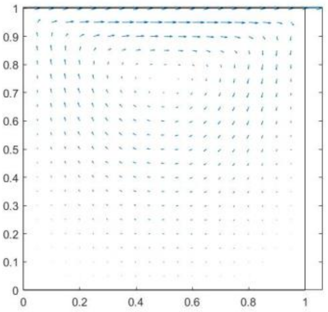


Figure 10. Q2Q1 Velocity Vectors

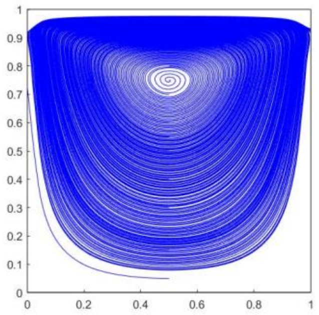


Figure 11. Q2Q1 Streamlines

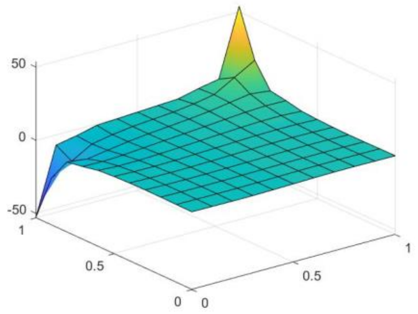


Figure 12. Q2Q1 Pressure Field

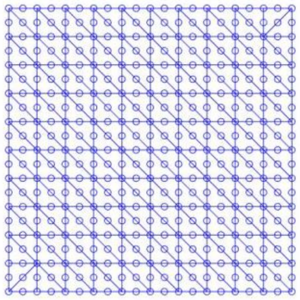


Figure 13. P2P1 Element scheme

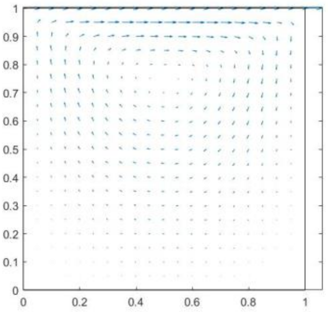


Figure 14. P2P1 Velocity Vectors

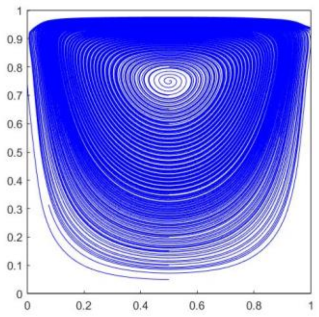


Figure 15. P2P1 Streamlines

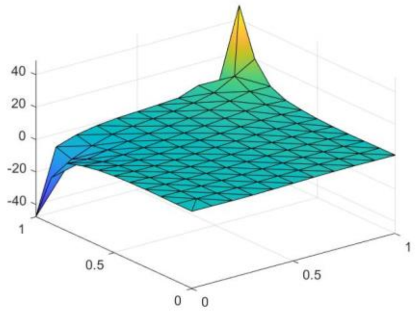


Figure 16. P2P1 Pressure Field

Both Q2Q1 and P2P1 elements give stable solution. The pressure field has a smooth trend across the entire field. And the streamlines are smooth and continuous.

Stabilization for Q1Q1 and P1P1 elements

3.3 Stabilization for Q1Q1 and P1P1 elements

We consider the GLS method to stabilize the formulation Q1Q1 and P1P1 to overcome the element to element oscillations. The stabilized weak form of equation is as below:

$$\begin{cases} a\left(w^h, v^h\right) + b\left(w^h, p^h\right) = \left(w^h, b^h\right) & for\ all\ w^h \in \boldsymbol{v^h} \\ b\left(v^h, q^h\right) + \tau\left(\nabla\, q^h, \nabla\, p^h - b^h\right) = 0 & for\ all\ q^h \in \boldsymbol{Q}^h \end{cases}$$

As concerns the value of the stabilization parameter,

$$\tau_e = \alpha_0 \frac{h_e^2}{4v}\ \ ,\ where\ \alpha_0 = \frac{1}{3}, v = 1\ and\ h_e = 0.1$$

Where h_e is the element length. v is viscosity which can be neglected in Stokes formulation.

3.3.1 Stabilized Q1Q1 Element and P1P1 Element

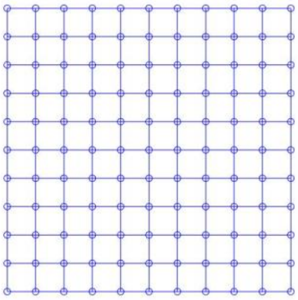


Figure 17. Stab Q1Q1 Element scheme

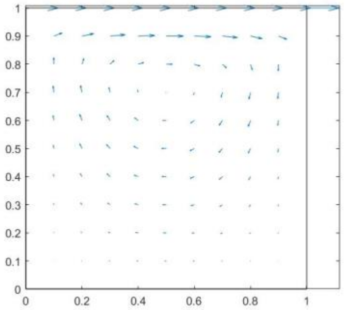


Figure 18. Stab Q1Q1 Velocity Vectors

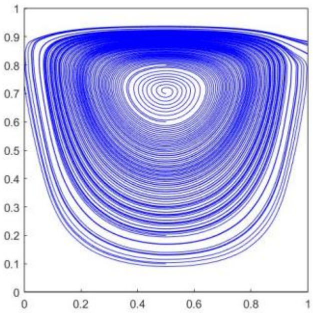


Figure 19. Stab Q1Q1 Streamlines

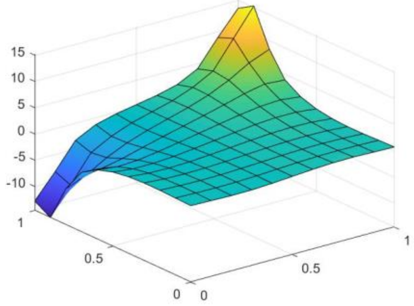


Figure 20. Stab Q1Q1 Pressure Field

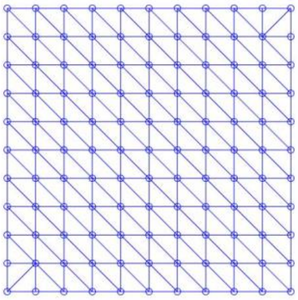


Figure 21. Stab P1P1 Element scheme

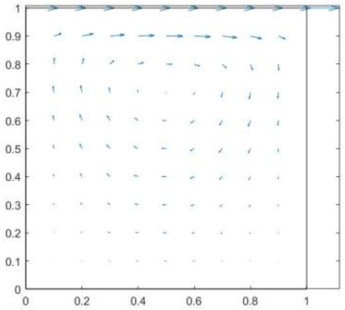


Figure 22. Stab P1P1 Velocity Vectors

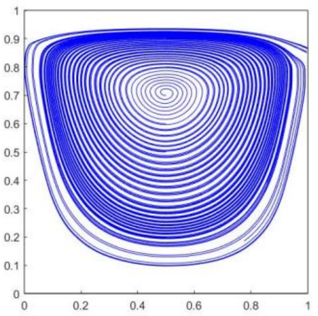


Figure 23. Stab P1P1 Streamlines

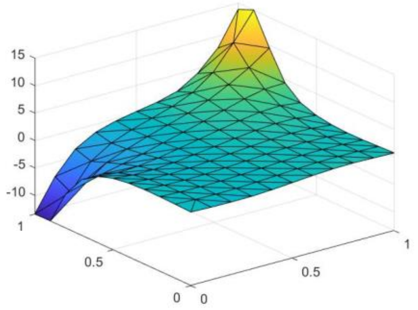


Figure 24. Stab P1P1 Pressure Field

We can find that now the pressure field becomes smooth and all the oscillations are vanished. So GLS stabilized scheme produce stable solution of stokes equation for both Q1Q1 and P1P1.

The equation for the GLS method is discretized as following:

$$\int_{\Omega^e} \tau L(w)\cdot R\left(\vec{V},p\right)d\Omega$$

Where $R = -\frac{1}{\rho}\nabla P + \frac{\mu}{\rho}\nabla^2\vec{V} + \vec{f}$

$L(w) = -\frac{1}{\rho}\nabla w_c + \frac{\mu}{\rho}\nabla^2\boldsymbol{w}$, where $\boldsymbol{w}$ is a combination of w_x and w_y

The GLS contribution to the continuity equation reads as following:

$$\int_{\Omega^e} \tau(\nabla w_c)\cdot(-\mu\nabla^2\vec{V} + \nabla p - \rho\vec{f})d\Omega$$

If the velocity approximation over the element is linear, the GLS will lead to

$$\int_{\Omega^e} \tau(\nabla w_c)\cdot(\nabla p - \rho\vec{f})d\Omega$$

Therefore, the modified weak form of the continuity equation is:

$$\int_{\Omega^e}\left[(-w_c)\left(\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}\right) - \tau\left(\frac{\partial w_c}{\partial x}\frac{\partial p}{\partial x}+\frac{\partial w_c}{\partial y}\frac{\partial p}{\partial y}\right)\right]d\Omega$$
$$= \int_{\Omega^e} -\tau\, w_c\left(\frac{\partial f_x}{\partial x}+\frac{\partial f_y}{\partial y}\right)d\Omega$$

Minus term is added to the continuity equation on purpose in order to get a symmetric stiffness matrix at the end.

The extra terms added to the continuity eq. are:

$$K_{ij} = \int_{\Omega^e} -\tau\left(\frac{\partial \hat{S}_i}{\partial x}\frac{\partial \hat{S}_j}{\partial x}+\frac{\partial \hat{S}_i}{\partial y}\frac{\partial \hat{S}_j}{\partial y}\right)d\Omega$$

$$F_i = \int_{\Omega^e} -\tau\hat{S}_i\left(\frac{\partial f_x}{\partial x}+\frac{\partial f_y}{\partial y}\right)d\Omega$$

Where $w_c = \hat{S}_i$

Being, $p^h(x,y) = \sum_{j=1}^{nenp} p_j\hat{S}_j(x,y)$

4. REFERENCE

[1] Lecture slides in Finite elements in fluid.

[2] Finite Element Methods for Flow Problems, Jean Donea and Antonio Huerta.

5. APPENDIX

GLS stabilization codes

```
function [K,G,f,L,f_q] = StokesSystemStable(X,T,XP,TP,referenceElement)
% [K,G,f] = StokesSystem(X,T,XP,TP,referenceElement)
% Matrices K, G and r.h.s vector f obtained after discretizing a Stokes problem
%
% X,T: nodal coordinates and connectivities for velocity
% XP,TP: nodal coordinates and connectivities for pressure
% referenceElement: reference element properties (quadrature, shape functions...)


elem = referenceElement.elemV;
ngaus = referenceElement.ngaus;
wgp = referenceElement.GaussWeights;
N = referenceElement.N;
Nxi = referenceElement.Nxi;
Neta = referenceElement.Neta;
NP = referenceElement.NP;
ngeom = referenceElement.ngeom;
mu=1;
h=XP(2)-XP(1);
tau1 = 1/3*h^2/(4*mu);


% Number of elements and number of nodes in each element
[nElem,nenV] = size(T);
nenP = size(TP,2);

% Number of nodes
nPt_V = size(X,1);
if elem == 11
    nPt_V = nPt_V + nElem;
end
nPt_P = size(XP,1);

% Number of degrees of freedom
nedofV = 2*nenV;
nedofP = nenP;
ndofV = 2*nPt_V;
ndofP = nPt_P;

K = zeros(ndofV,ndofV);
G = zeros(ndofP,ndofV);
f = zeros(ndofV,1);
L = zeros(ndofP,ndofP);
f_q=zeros(ndofP,1);

% Loop on elements
for ielem = 1:nElem
    % Global number of the nodes in element ielem
    Te = T(ielem,:);
    TPe = TP(ielem,:);
    % Coordinates of the nodes in element ielem
    Xe = X(Te(1:ngeom),:);
    % Degrees of freedom in element ielem
    Te_dof = reshape([2*Te-1; 2*Te],1,nedofV);
    TPe_dof = TPe;

    % Element matrices
    [Ke,Ge,fe,Le,f_qe] =
EleMatStokes(Xe,ngeom,nedofV,nedofP,ngaus,wgp,N,Nxi,Neta,NP,tau1);

    % Assemble the element matrices
    K(Te_dof, Te_dof) = K(Te_dof, Te_dof) + Ke;
    G(TPe_dof,Te_dof) = G(TPe_dof,Te_dof) + Ge;
    f(Te_dof) = f(Te_dof) + fe;
    L(TPe_dof,TPe_dof)= L(TPe_dof,TPe_dof) + Le;
    f_q(TPe_dof) = f_q(TPe_dof) + f_qe;
end






function [Ke,Ge,fe,Le,f_qe] =
EleMatStokes(Xe,ngeom,nedofV,nedofP,ngaus,wgp,N,Nxi,Neta,NP,tau1)
%

Ke = zeros(nedofV,nedofV);
Ge = zeros(nedofP,nedofV);
fe = zeros(nedofV,1);
Le = zeros(nedofP,nedofP);
f_qe=zeros(nedofP,1);


% Loop on Gauss points
for ig = 1:ngaus
    N_ig    = N(ig,:);
    Nxi_ig  = Nxi(ig,:);
    Neta_ig = Neta(ig,:);
    NP_ig = NP(ig,:);
    Jacob = [
        Nxi_ig(1:ngeom)*(Xe(:,1))   Nxi_ig(1:ngeom)*(Xe(:,2))
        Neta_ig(1:ngeom)*(Xe(:,1))  Neta_ig(1:ngeom)*(Xe(:,2))
        ];
    dvolu = wgp(ig)*det(Jacob);
    res = Jacob\[Nxi_ig;Neta_ig];
    nx = res(1,:);
    ny = res(2,:);

    Ngp = [reshape([1;0]*N_ig,1,nedofV); reshape([0;1]*N_ig,1,nedofV)];
    % Gradient
    Nx = [reshape([1;0]*nx,1,nedofV); reshape([0;1]*nx,1,nedofV)];
    Ny = [reshape([1;0]*ny,1,nedofV); reshape([0;1]*ny,1,nedofV)];
    %NPx = [reshape([1;0]*nx,1,nedofP); reshape([0;1]*nx,1,nedofP)];
    %NPy = [reshape([1;0]*ny,1,nedofP); reshape([0;1]*ny,1,nedofP)];
    % Divergence
    dN = reshape(res,1,nedofV);

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