HOMEWORK STEADY CONVECTION

-FINITE ELEMENTS IN FLUIDS-

Marcos Boniquet

5 METHODS IMPLEMENTED, CONSIDERING STABILIZATION PARAMETERS .

CONVECTION-DIFFUSION equation:

$$a\, u_x - \boldsymbol{\nu}^*\, u_{xx} = f \quad x \in (0,1)$$
$$U(0)=U_0$$
$$U(1)=U_1$$

GALERKIN

$K_e = \int (N a N_x + N_x \boldsymbol{\nu} N_x)\, d\Omega$
$f_e = \int N s\, d\Omega$

SU

Stabilization term, adding an artificial convective term to erase oscillations.

$K_e = \int (N a N_x + N_x (\boldsymbol{\nu} + \bar{u}) N_x)\, d\Omega$
$f_e = \int N s\, d\Omega$

SUPG

More accurate.

$$\mathcal{P}(w) = \mathcal{L}(w) = \boldsymbol{a} \cdot \boldsymbol{\nabla} w$$

$K_e = \int (N a N_x + N_x \boldsymbol{\nu} N_x + \tau a^2 N_x N_x)\, d\Omega$ if $\tau = \bar{u}/a \rightarrow$ SU

$\tau a^2 N_x N_x$ = stabilization term

$f_e = \int N s\, d\Omega$

In other words:
$K_e = \int (N a N_x + N_x (\boldsymbol{\nu} + \tau a^2) N_x)\, d\Omega$ (if $\tau = \bar{u}/a \rightarrow$ SU)
$f_e = \int N s\, d\Omega$

GLS

If linear and NO reaction is SUPG.

$$\mathcal{P}(w) = \mathcal{L}(w) = \boldsymbol{a} \cdot \boldsymbol{\nabla} w - \boldsymbol{\nabla} \cdot (\nu \boldsymbol{\nabla} w) + \sigma w$$

Considering+quadratic:

$K_e=\int(N a N_x+N_x \boldsymbol{\nu} N_x-(\tau a^2 N_x N_x+a\boldsymbol{\nu} N_x N_{xx}-a\boldsymbol{\nu} N_{xx} N_x+\boldsymbol{\nu}^2 N_{xx} N_{xx})\, d\Omega$

$\tau a^2 N_x N_x+a\boldsymbol{\nu} N_x N_{xx}-a\boldsymbol{\nu} N_{xx} N_x+\boldsymbol{\nu}^2 N_{xx} N_{xx}$= stabilization term
$a\boldsymbol{\nu} N_x N_{xx}-a\boldsymbol{\nu} N_{xx} N_x+\boldsymbol{\nu}^2 N_{xx} N_{xx}$=quadratic term

$f_e=\int Ns\, d\Omega$

Considering reaction+quadratic:

$K_e=\int(N a N_x+N_x \boldsymbol{\nu} N_x-(\tau a^2 N_x N_x+a\boldsymbol{\nu} N_x N_{xx}-a\boldsymbol{\nu} N_{xx} N_x+\boldsymbol{\nu}^2 N_{xx} N_{xx}+u\sigma N_x N-\boldsymbol{\nu}\sigma N_{xx} N+\sigma a N N_x-\boldsymbol{\nu}\sigma N N_{xx}+\sigma N N)\, d\Omega$

stabilization terrm =
$\tau a^2 N_x N_x+a\boldsymbol{\nu} N_x N_{xx}-a\boldsymbol{\nu} N_{xx} N_x+\boldsymbol{\nu}^2 N_{xx} N_{xx}+u\sigma N_x N-\boldsymbol{\nu}\sigma N_{xx} N+\sigma a N N_x-\boldsymbol{\nu}\sigma N N_{xx}+\sigma N N$
quadratic term= $a\boldsymbol{\nu} N_x N_{xx}-a\boldsymbol{\nu} N_{xx} N_x+\boldsymbol{\nu}^2 N_{xx} N_{xx}$
reactive term=$u\sigma N_x N-\boldsymbol{\nu}\sigma N_{xx} N+\sigma a N N_x-\boldsymbol{\nu}\sigma N N_{xx}+\sigma N N$

$f_e=\int(Ns + aNx-\boldsymbol{\nu}\bar{u} N_{xx}+\sigma N)d\Omega$

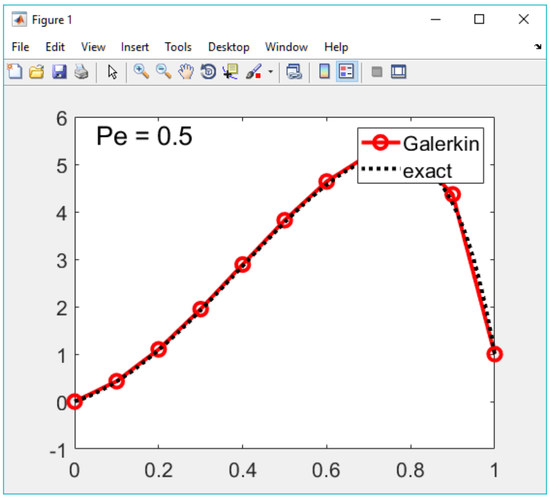
SGS

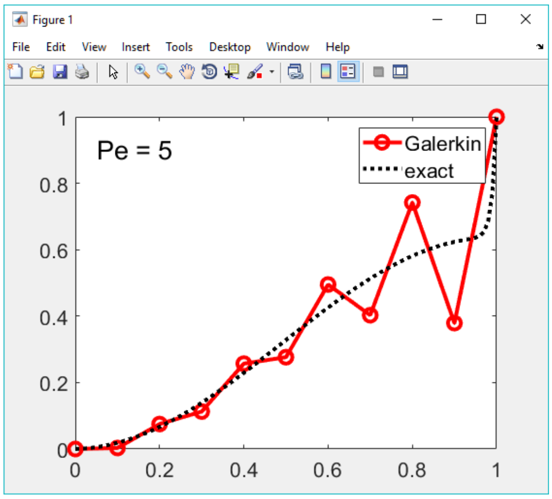
$$\mathcal{P}(w)=\mathcal{L}(w)=\boldsymbol{a}\cdot\boldsymbol{\nabla} w+\boldsymbol{\nabla}\cdot(\nu\boldsymbol{\nabla} w)-\ \sigma w$$

Analogously.

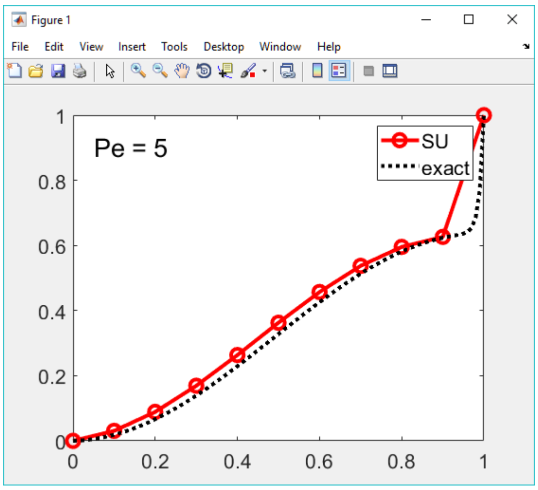
First methods are not worth be commented in depth, with the exception of Pe number being important for Galerkin, because if Pe<5 it can be unstable. SU method erases oscillations and SUPG, given a precise value of tau, can be *extremely* precise for a linear non-reactive system.

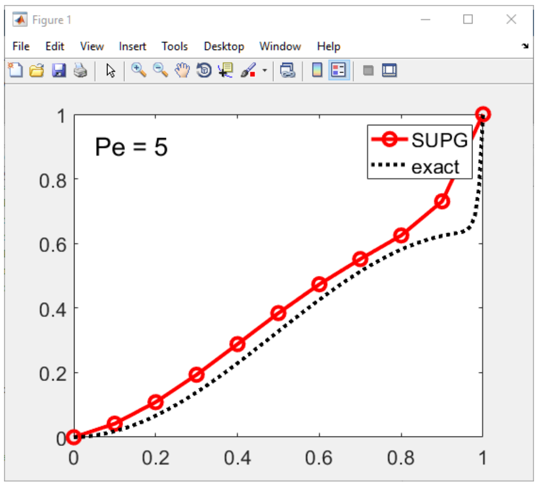
Pe Number affecting Galerkin Method:





SU & SUPG, the latter more accurate if tau value is adequate:



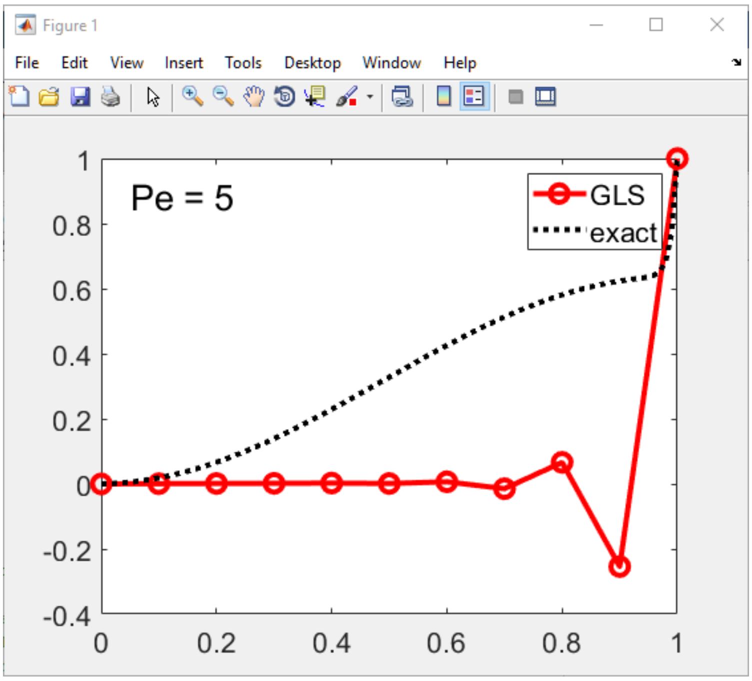


SUPG $\tau = h*(1+9/Pe^2)^{-1/2}$

We center the efforts on **GLS**.
The Matlab code has been implemented to be capable of computing with **quadratic elements** too. (p=2). A set of T2 coordinate-of-the-elements matrix has been assembled, and also has been considered possibility of non-zero reaction, being transformed the function **GLS** as following:

```
function [K,f] = GLS_system (X,T,T2,referenceElement,example)

[...]

K2 = zeros(nPt*2-1,nPt*2-1);
f2 = zeros(nPt*2-1,1);

% Loop on elements
for ielem = 1:nElem
   Te = T(ielem,:);
   Te2= T2(ielem,:);
   Xe = X(Te);
   Xe2=[Xe(1),(Xe(2)+Xe(1))/2,Xe(2)]';
   h = Xe(end) - Xe(1);
   Ke = zeros(nen);
   fe = zeros(nen,1);
   % Loop on Gauss points
   for ig = 1:ngaus
      N_ig = N(ig,:);
      Nx_ig = Nxi(ig,:)*2/h;
      Nxx_ig = Nxxi(ig,:)*2/h;      %revisar
      w_ig = wgp(ig)*h/2;           %revisar  incluye jacobiano, que cambia para cada p

      Ke = Ke + w_ig*(N_ig'*a*Nx_ig + Nx_ig'*nu*Nx_ig...
         + tau*(a^2)*Nx_ig'*Nx_ig +...% +     -nu*a*Nx_ig'*Nxx_ig  + ...
         a*sigma*Nx_ig'*N_ig + ...
         a*nu*Nxx_ig'*Nx_ig  +  (nu^2)*Nxx_ig'*Nxx_ig  -    nu*sigma*Nxx_ig'*N_ig+    ...
         sigma*a*N_ig'*Nx_ig  -  sigma*nu*N_ig'*Nxx_ig   +   (sigma^2)*N_ig'*N_ig ...
         + sigma*N_ig'*N_ig  );

      if p==1
      x = N_ig*Xe; % x-coordinate of the gauss point!!!!!!!  f en Gauss points!!!!!    es un procucto escalar!

      elseif p==2
      x = N_ig*Xe2; % x-coordinate of the gauss point!!!!!!!  f en Gauss points!!!!!    es un procucto escalar!
      end

      s = SourceTerm(x,example);
      fe = fe + w_ig*(N_ig)'*s;
        % (a*Nx_ig'
      %  -(nu*Nxx_ig'+sigma*N_ig')*tau*s;
   end
   % Assembly
   if p==1
   K(Te,Te) = K(Te,Te) + Ke;
   f(Te) = f(Te) + fe;
   elseif p==2
   K2(Te2,Te2) = K2(Te2,Te2) + Ke;
   f2(Te2) = f2(Te2) + fe;
   end
end
```

Of course, if p=1 (linear) but considered a σ≠0, the results differ from the exact solution given that the exact solution does not include this casuistry, is only **convective**.



*GLS σ=20 (**EXACT SOLUTION W/O REACTION**)*