Finite Elements in Fluids

Nora Wieczorek i Masdeu

February 2019

We consider the Steady transport problem with **strong form**

$$\mathbf{a} \cdot \nabla u - \nabla \cdot (\nu \nabla u) + \sigma u = s \qquad \text{in } \Omega \tag{1}$$

with Dirichlet boundary conditions $u = u_D$ in $\delta\Omega$.

1 Weak form derivation

Let $\omega \in H_1(\Omega)$ such that $\omega = 0$ on $\delta\Omega$. Then,

$$\int_\Omega \omega(\mathbf{a} \cdot \nabla u) d\Omega - \int_\Omega \omega(\nabla \cdot (\nu \nabla u)) d\Omega + \int_\Omega \omega(\sigma u) d\Omega = \int_\Omega \omega s d\Omega \tag{2}$$

Using that $\nabla \cdot (\omega(\nu \nabla u)) = \nabla\omega \cdot (\nu \nabla u) + \omega(\nabla \cdot (\nu \nabla u))$ and the divergence theorem ($\int_\Omega \nabla \cdot \mathbf{F} = \int_{\delta\Omega} \mathbf{F} \cdot \mathbf{n} d\delta\Omega$) the diffusion term turns into,

$$-\int_\Omega \omega(\nabla \cdot (\nu \nabla u)) d\Omega = \int_\Omega \nabla\omega \cdot (\nu \nabla u) d\Omega - \int_{\delta\Omega} \omega(\nu \nabla u) \cdot \mathbf{n} d\delta\Omega \tag{3}$$

As we had considered that $\omega = 0$ on $\delta\Omega$, the second term disapears.

Thus, the **weak form of the problem** is:

Find $u \in H_1(\Omega)$ s.t. $u = u_D$ on $\delta\Omega$, such that

$$\int_\Omega \omega(\mathbf{a} \cdot \nabla u) d\Omega + \int_\Omega \nabla\omega \cdot (\nu \nabla u) d\Omega + \int_\Omega \omega(\sigma u) d\Omega = \int_\Omega \omega s d\Omega \tag{4}$$

$\forall \omega \in H_1(\Omega)$ s.t. $\omega = 0$ on $\delta\Omega$.

1.1 Streamline upwind (SU)

To solve convection dominated problems ($Pe > 1$), we add artificial diffusion $\bar{\nu}$, so we end up with the following equation

$$\int_\Omega \omega(\mathbf{a} \cdot \nabla u) d\Omega + \int_\Omega \nabla\omega \cdot (\nu \nabla u) d\Omega + \int_\Omega \omega(\sigma u) d\Omega + \int_\Omega \frac{\bar{\nu}}{||\mathbf{a}||^2} (\mathbf{a} \cdot \nabla\omega)(\mathbf{a} \cdot \nabla u) d\Omega = \int_\Omega \omega s d\Omega \tag{5}$$

1.2 Consistent stabilized formulations

Our PDE can be written as $\mathcal{R}(u) = 0$ with

$$\mathcal{R}(u) = \mathbf{a} \cdot \nabla u - \nabla \cdot (\nu \nabla u) + \sigma u - s = \mathcal{L}(u) - s \tag{6}$$

Another stabilization technique is to add a term that is a multiple of by this residual, i.e.

$$\sum_e \int_{\Omega_e} \mathcal{P}(\omega) \tau \mathcal{R}(u) d\Omega \tag{7}$$

This new formulation gives us the same solution as the one without, so we can say that is a consistent formulation.

1.2.1 Streamline upwind Petrov-Galerkin (SUPG)

We consider a consistent stabilized formulation using $\mathcal{P}(\omega) = \mathbf{a} \cdot \nabla \omega$.

So the new weak form is

$$\begin{aligned} & \int_\Omega \omega (\mathbf{a} \cdot \nabla u) d\Omega + \int_\Omega \nabla \omega \cdot (\nu \nabla u) d\Omega + \int_\Omega \omega (\sigma u) d\Omega \\ + \quad & \tau \sum_e \int_{\Omega_e} \left[(\mathbf{a} \cdot \nabla \omega) (\mathbf{a} \cdot \nabla u) - \nu (\mathbf{a} \cdot \nabla \omega) (\nabla \cdot \nabla u) + \sigma (\mathbf{a} \cdot \nabla \omega) u \right] d\Omega \\ = \quad & \int_\Omega \omega s d\Omega + \tau \sum_e \int_{\Omega_e} s (\mathbf{a} \cdot \nabla \omega) d\Omega \end{aligned}$$

1.2.2 Galerkin least-squares (GLS)

Is a consistent stabilized formulation using $\mathcal{P}(\omega) = \mathcal{L}(\omega) = \mathbf{a} \cdot \nabla \omega - \nabla \cdot (\nu \nabla \omega) + \sigma \omega$.

So the new weak form is

$$\begin{aligned} & \int_\Omega \omega (\mathbf{a} \cdot \nabla u) d\Omega + \int_\Omega \nabla \omega \cdot (\nu \nabla u) d\Omega + \int_\Omega \omega (\sigma u) d\Omega \\ + \quad & \tau \sum_e \int_{\Omega_e} \left[(\mathbf{a} \cdot \nabla \omega) (\mathbf{a} \cdot \nabla u) - \nu (\mathbf{a} \cdot \nabla \omega) (\nabla \cdot \nabla u) + \sigma (\mathbf{a} \cdot \nabla \omega) u \right] d\Omega \\ - \quad & \tau \nu \sum_e \int_{\Omega_e} \left[(\nabla \cdot \nabla \omega) (\mathbf{a} \cdot \nabla u) - \nu (\nabla \cdot \nabla \omega) (\nabla \cdot \nabla u) - \sigma (\nabla \omega) \cdot (\nabla u) \right] d\Omega \\ + \quad & \tau \sigma \sum_e \int_{\Omega_e} \left[\omega (\mathbf{a} \cdot \nabla u) + \nu (\nabla \omega) \cdot (\nabla u) + \sigma \omega u \right] d\Omega \\ - \quad & \tau \sigma \nu \sum_e \int_{\delta \Omega_e} \left[(\nabla \omega) u \cdot \mathbf{n} + \omega (\nabla u) \cdot \mathbf{n} \right] d\delta\Omega \\ = \quad & \int_\Omega \omega s d\Omega + \tau \sum_e \int_{\Omega_e} s \left[(\mathbf{a} \cdot \nabla \omega) - \nu (\nabla \cdot \nabla \omega) + \sigma \omega \right] d\Omega \end{aligned}$$

2 Finite Elements Discretization

Let us define a mesh $\{\Omega_e\}_{e=1,...,n_{Elem}}$ such that $\cup\Omega_e = \Omega$, where the nodes are the points with coordinates $\{\mathbf{x_i}\}_{i=1,...,n_{Nodes}}$.

We will take as a basis functions of $H_1(\Omega)$ the set $\{N_i(\mathbf{x})\}_{i=1,...,n_{Nodes}}$ such that $N_i(\mathbf{x_j}) = \delta_{ij}$. NOTE: we would consider polynomial functions of different degrees to obtain different order approximations.

We will define the FEM approximation of our function as a linear combination of the basis functions

$$u(\mathbf{x}) \approx u^h(\mathbf{x}) = \sum_{i=1}^{n_{Nodes}} u_i N_i(\mathbf{x}) \tag{8}$$

We will do the same for ω,

$$\omega(\mathbf{x}) = \sum_{i=1}^{n_{Nodes}} \omega_i N_i(\mathbf{x}) \tag{9}$$

Imposing 8 and 9 to the weak form, we obtain

$$\begin{aligned} &\sum_{i,j=1}^{n_{Nodes}} \omega_i u_j \int_\Omega N_i(\mathbf{x}) \left(\mathbf{a} \cdot \nabla N_j(\mathbf{x})\right) d\Omega + \nu \sum_{i,j=1}^{n_{Nodes}} \omega_i u_j \int_\Omega \left(\nabla N_i(\mathbf{x})\right)^T \cdot \nabla N_j(\mathbf{x}) d\Omega \\ + \quad &\sigma \sum_{i,j=1}^{n_{Nodes}} \omega_i u_j \int_\Omega N_i(\mathbf{x}) N_j(\mathbf{x}) d\Omega = \sum_{i=1}^{n_{Nodes}} \omega_i \int_\Omega s N_i(\mathbf{x}) d\Omega \end{aligned} \tag{10}$$

Considering the vectors $\mathbf{W} = (\omega_1, ..., \omega_{n_{Nodes}})^T$, $\mathbf{U} = (u_1, ..., u_{n_{Nodes}})^T$, $\mathbf{F} = \left(\int_\Omega s N_i(\mathbf{x}) d\Omega\right)_{i=1,...,n_{Nodes}}$ and the matrices

$$L = \left(\int_\Omega N_i(\mathbf{a} \cdot \nabla N_j) d\Omega\right)_{i,j}, \quad K = \left(\int_\Omega \left(\nabla N_i\right)^T \cdot \nabla N_j d\Omega\right)_{i,j}, \quad M = \left(\int_\Omega N_i N_j d\Omega\right)_{i,j}, \quad A = L + \nu K + \sigma M$$

Using this notation we can write 10, as the following equation:

$$\mathbf{W}^T A \mathbf{U} = \mathbf{W}^T \mathbf{F} \tag{11}$$

As we had imposed $\omega = 0$ on $\delta\Omega$, we know that if $x_i \in \delta\Omega \Rightarrow \omega_i = 0$. So we will discard the rows of our system that coincide with the index of the boundary nodes. As 11 has to be accomplished $\forall \omega \in H_1(\Omega)$, finding the solution is equivalent to solve the following linear system

$$(A)_{i \text{ s.t. } x_i \notin \delta\Omega, j=1,...,n_{Nodes}} \mathbf{U} = (\mathbf{F})_{i \text{ s.t. } x_i \notin \delta\Omega} \tag{12}$$

We also know the nodal values $u_j = u_D(\mathbf{x_j})$ for $x_j \in \delta\Omega$. Thus, we end up with the following system

$$(A)_{i \text{ s.t. } x_i \notin \delta\Omega, j \text{ s.t. } x_j \notin \delta\Omega} \mathbf{U}_{j \text{ s.t. } x_j \notin \delta\Omega} = (\mathbf{F})_{i \text{ s.t. } x_i \notin \delta\Omega} - (A)_{i \text{ s.t. } x_i \notin \delta\Omega, j \text{ s.t. } x_j \in \delta\Omega} \mathbf{U_D} \tag{13}$$

2.1 Discretization for the Streamline Upwind (SU) method

We add the artificial diffusion to equation 10,

$$\begin{aligned} & \sum_{i,j=1}^{n_{Nodes}} \omega_i u_j \int_\Omega N_i(\mathbf{x}) \left(\mathbf{a} \cdot \nabla N_j(\mathbf{x})\right) d\Omega + \nu \sum_{i,j=1}^{n_{Nodes}} \omega_i u_j \int_\Omega \left(\nabla N_i(\mathbf{x})\right)^T \cdot \nabla N_j(\mathbf{x}) d\Omega \\ + \quad & \sigma \sum_{i,j=1}^{n_{Nodes}} \omega_i u_j \int_\Omega N_i(\mathbf{x}) N_j(\mathbf{x}) d\Omega + \bar{\nu} \sum_{i,j=1}^{n_{Nodes}} \omega_i u_j \int_\Omega (\nabla N_i(\mathbf{x}))^T \nabla N_j(\mathbf{x}) d\Omega \\ = \quad & \sum_{i=1}^{n_{Nodes}} \omega_i \int_\Omega s N_i(\mathbf{x}) d\Omega \end{aligned} \tag{14}$$

So defining $A_{SU} = L + (\nu + \bar{\nu})K + \sigma M$ and applying the same that we had applied before, we end up into having to solve the following linear system

$$(A_{SU})_{i \text{ s.t. } x_i \notin \delta\Omega, j \text{ s.t. } x_j \notin \delta\Omega} \mathbf{U}_{j \text{ s.t. } x_j \notin \delta\Omega} = (\mathbf{F})_{i \text{ s.t. } x_i \notin \delta\Omega} - (A_{SU})_{i \text{ s.t. } x_i \notin \delta\Omega, j \text{ s.t. } x_j \in \delta\Omega} \mathbf{U_D} \tag{15}$$

2.2 Discretization for the Streamline Upwind Petrov-Galerkin (SUPG) method

Considering equation 8 and susbtituting ω and u^h we have

$$\begin{aligned} & \sum_{i,j=1}^{n_{Nodes}} \omega_i u_j \int_\Omega N_i(\mathbf{x}) \left(\mathbf{a} \cdot \nabla N_j(\mathbf{x})\right) d\Omega + \nu \sum_{i,j=1}^{n_{Nodes}} \omega_i u_j \int_\Omega \left(\nabla N_i(\mathbf{x})\right)^T \cdot \nabla N_j(\mathbf{x}) d\Omega + \sigma \sum_{i,j=1}^{n_{Nodes}} \omega_i u_j \int_\Omega N_i(\mathbf{x}) N_j(\mathbf{x}) d\Omega \\ + \quad & \tau \sum_{i,j=1}^{n_{Nodes}} \omega_i u_j \sum_e \left[||a||^2 \int_{\Omega_e} (\nabla N_i(\mathbf{x}))^T \nabla N_j(\mathbf{x}) d\Omega - \nu \int_{\Omega_e} (\mathbf{a} \cdot \nabla N_i(\mathbf{x})) \nabla \cdot \nabla N_j(\mathbf{x}) d\Omega + \sigma \int_{\Omega_e} (\mathbf{a} \cdot \nabla N_i(\mathbf{x})) N_j(\mathbf{x}) d\Omega \right] \\ = \quad & \sum_{i=1}^{n_{Nodes}} \omega_i \int_\Omega s N_i(\mathbf{x}) d\Omega + \tau \sum_{i=1}^{n_{Nodes}} \omega_i \sum_e \int_{\Omega_e} s \left(\mathbf{a} \cdot \nabla N_i(\mathbf{x})\right) d\Omega \end{aligned} \tag{16}$$

So defining $B = \left(\int_\Omega (\mathbf{a} \cdot \nabla N_i(\mathbf{x})) \nabla \cdot \nabla N_j(\mathbf{x}) d\Omega\right)_{i,j}$, $\mathbf{G} = (\int_\Omega s\left(\mathbf{a} \cdot \nabla N_i(\mathbf{x})\right) d\Omega)_i$ and $A_{SUPG} = L + (\nu + \tau||\mathbf{a}||^2)K + \sigma M - \tau\nu B$, and applying the same that we had applied in the Galerkin case, we end up into having to solve the following linear system

$$(A_{SUPG})_{i \text{ s.t. } x_i \notin \delta\Omega, j \text{ s.t. } x_j \notin \delta\Omega} \mathbf{U}_{j \text{ s.t. } x_j \notin \delta\Omega} = (\mathbf{F} + \tau\mathbf{G})_{i \text{ s.t. } x_i \notin \delta\Omega} - (A_{SUPG})_{i \text{ s.t. } x_i \notin \delta\Omega, j \text{ s.t. } x_j \in \delta\Omega} \mathbf{U_D}$$

2.3 Discretization for the Galerkin least-squares (GLS) method

Considering equation 8 and susbtituting ω and u^h we have

$$
\begin{aligned}
& \sum_{i,j=1}^{n_{Nodes}} \omega_i u_j \int_\Omega N_i(\mathbf{x}) \left(\mathbf{a}^T \cdot \nabla N_j(\mathbf{x})\right) d\Omega + \nu \sum_{i,j=1}^{n_{Nodes}} \omega_i u_j \int_\Omega \left(\nabla N_i(\mathbf{x})\right)^T \cdot \nabla N_j(\mathbf{x}) d\Omega + \sigma \sum_{i,j=1}^{n_{Nodes}} \omega_i u_j \int_\Omega N_i(\mathbf{x}) N_j(\mathbf{x}) d\Omega \\
+ \quad & \tau \sum_{i,j=1}^{n_{Nodes}} \omega_i u_j \sum_e \left[||a||^2 \int_{\Omega_e} (\nabla N_i(\mathbf{x}))^T \nabla N_j(\mathbf{x}) d\Omega - \nu \int_{\Omega_e} (\mathbf{a} \cdot \nabla N_i(\mathbf{x}))(\nabla \cdot \nabla N_j(\mathbf{x})) d\Omega + \sigma \int_{\Omega_e} (\mathbf{a} \cdot \nabla N_i(\mathbf{x})) N_j(\mathbf{x}) d\Omega \right] \\
- \quad & \tau\nu \sum_{i,j=1}^{n_{Nodes}} \omega_i u_j \sum_e \Bigg[\int_{\Omega_e} (\nabla \cdot \nabla N_i(\mathbf{x}))(\mathbf{a} \cdot \nabla N_j(\mathbf{x})) d\Omega - \nu \int_{\Omega_e} (\nabla \cdot \nabla N_i(\mathbf{x}))(\nabla \cdot \nabla N_j(\mathbf{x})) d\Omega - \sigma \int_{\Omega_e} (\nabla N_i(\mathbf{x}))^T \cdot \nabla N_j(\mathbf{x}) d \\
+ \quad & \tau\sigma \sum_{i,j=1}^{n_{Nodes}} \omega_i u_j \sum_e \left[\int_{\Omega_e} N_i(\mathbf{x})(\mathbf{a} \cdot \nabla N_j(\mathbf{x})) d\Omega + \nu \int_{\Omega_e} (\nabla N_i(\mathbf{x}))^T \nabla N_j(\mathbf{x}) d\Omega + \sigma \int_{\Omega_e} N_i(\mathbf{x}) N_j(\mathbf{x}) d\Omega \right] \\
- \quad & \tau\nu\sigma \sum_{i,j=1}^{n_{Nodes}} \omega_i u_j \sum_e \left[\int_{\delta\Omega_e} (\nabla N_i(\mathbf{x})) N_j(\mathbf{x}) \cdot \mathbf{n} d\delta\Omega + \int_{\delta\Omega_e} N_i(\mathbf{x})(\nabla N_j(\mathbf{x})) \cdot \mathbf{n} d\delta\Omega \right] \\
= \quad & \sum_{i=1}^{n_{Nodes}} \omega_i \int_\Omega s N_i(\mathbf{x}) d\Omega + \tau \sum_{i=1}^{n_{Nodes}} \omega_i \sum_e \left[\int_{\Omega_e} \left(\mathbf{a} \cdot \nabla N_i(\mathbf{x})\right) s d\Omega + \nu \int_{\Omega_e} (\nabla \cdot \nabla N_i(\mathbf{x})) s d\Omega + \sigma \int_{\Omega_e} N_i(\mathbf{x}) s d\Omega \right] \qquad (
\end{aligned}
$$

So defining $C = \left(\int_\Omega (\nabla \cdot \nabla N_i(\mathbf{x})) \nabla \cdot \nabla N_j(\mathbf{x}) d\Omega\right)_{i,j}$, $D = \left(\int_{\delta\Omega} (\nabla N_i(\mathbf{x})) N_j(\mathbf{x}) \cdot \mathbf{n} d\delta\Omega\right)_{i,j}$, $\mathbf{H} = (\int_\Omega s \left(\nabla \cdot \nabla N_i(\mathbf{x})\right) d\Omega)_i$ and

$$A_{GLS} = (1 + \tau\sigma)(L + \sigma M) + \tau\sigma L^T + (\nu + \tau||\mathbf{a}||^2 + 2\tau\sigma\nu)K + \tau\nu(B - B^T + \nu C - \sigma D - \sigma D^T) \tag{18}$$

and applying the same that we had applied, we end up into having to solve the following linear system

$$(A_{GLS})_{i \text{ s.t. } x_i \notin \delta\Omega, j \text{ s.t. } x_j \notin \delta\Omega} \mathbf{U}_{j \text{ s.t. } x_j \notin \delta\Omega} = (\mathbf{F} + \tau(\sigma\mathbf{F} + \mathbf{G} + \nu\mathbf{H}))_{i \text{ s.t. } x_i \notin \delta\Omega} - (A_{GLS})_{i \text{ s.t. } x_i \notin \delta\Omega, j \text{ s.t. } x_j \in \delta\Omega} \mathbf{U_D}$$

2.4 For a one dimensional domain

Let us consider $\Omega = [0, 1]$. We will discretize our domain using a n_{Elem} uniform mesh, i.e in the **first degree** case the nodes are located at $\{x_i\}_{i=0,...,n_{Elem}} = \{i/n_{Elem}\}_{i=0,...,n_{Elem}}$, so $n_{Nodes} = n_{Elem} + 1$, and the elements are defined by $\Omega_e = ((e-1)/n_{Elem}, e/n_{Elem})$ for $e = 1, ..., n_{Nodes}$. In the **second degree** case, the nodes are located at $\{x_i\}_{i=0,...,n_{Elem}} = \{i/n_{Elem}\}_{i=0,...,2n_{Elem}}$, so $n_{Nodes} = 2n_{Elem} + 1$, and the elements are defined by $\Omega_e = ((2e-2)/2n_{Elem}, (2e-1)/2n_{Elem}, (2e)/2n_{Elem})$.

3 Our problems

We consider the convection-diffusion equation, i.e. ($\sigma = 0$),

$$\mathbf{a} \cdot \nabla u - \nabla \cdot (\nu \nabla u) = s \qquad \text{in } \Omega \tag{19}$$

with

- **Problem 1:** $s = 0$ and boundary conditions u(0)= 0, u(1) = 1.
- **Problem 3:** $s = sin(\pi x)$ and boundary conditions u(0)= 0, u(1) = 1.
- **Problem 4:** $s = 10e^{-5x} - 4e^{-x}$ and boundary conditions u(0)= 0, u(1) = 1.

Thus, to apply every method we will have to solve the following linear system

$$(A)_{i,j=2,\ldots,n_{Nodes}-1} \mathbf{U}_{j=2,\ldots,n_{Nodes}-1} = \left(\bar{\mathbf{F}}\right)_{i=2,\ldots,n_{Nodes}-1} - (A)_{i=2,\ldots,n_{Nodes}-1,j\in\{1,n_{Nodes}\}} (\mathbf{0},\mathbf{1})^{\mathbf{T}} \tag{20}$$

having the following matrix A and vector $\bar{\mathbf{F}}$ for each method:

- **Galerkin:** $A = (L + \nu K)$ and $\bar{\mathbf{F}} = \mathbf{F}$.
- **SU:** $A = L + (\nu + \bar{\nu})K$ and $\bar{\mathbf{F}} = \mathbf{F}$.
- **SUPG:** $A = L + (\nu + \tau||\mathbf{a}||^2)K - \tau\nu B$ and $\bar{\mathbf{F}} = \mathbf{F} + \tau\mathbf{G}$.
- **GLS:** $A = L + (\nu + \tau||\mathbf{a}||^2)K + \tau\nu(B - B^T + \nu C)$ and $\bar{\mathbf{F}} = \mathbf{F} + \tau\mathbf{G} + \tau\nu\mathbf{H}$.

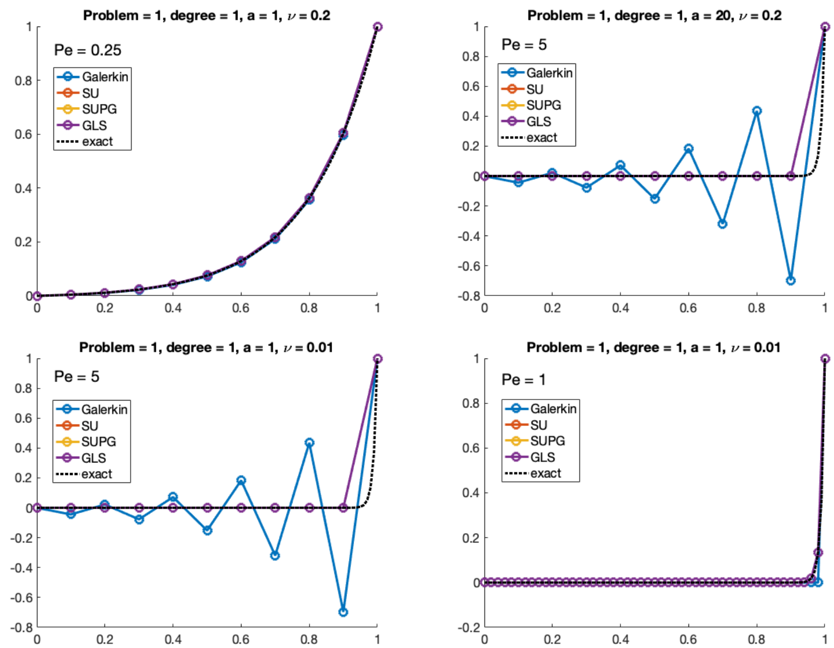

Figure 1: Problem 1 solved with linear elements

4 Results

To solve the one dimensional system we will use linear and quadratic elements.

4.1 Linear Elements

Note that if N_i are linear functions, $\nabla \cdot \nabla N_i = 0$, so $B, C = 0$ and $\mathbf{H} = 0$. Thus solving the problems with linear functions using the SUPG formulation is equivalent to solve them with linear functions using the GLS formulation as we can see in Figures 1, 2 and 3.

We can also see that oscillations appear when $Pe > 1$, and that they are solved for the three methods. We can also see in Figures 2 and 3 that the SU formulation is non consitent.

4.2 Quadratic Elements

For the quadratic elements, $\nabla \cdot \nabla N_i \neq 0$, so SUPG and GLS formulation gave us different solutions have different systems to solve, but as we can see in Figures 4, 5 and 6, they gave us the same approximate solution.

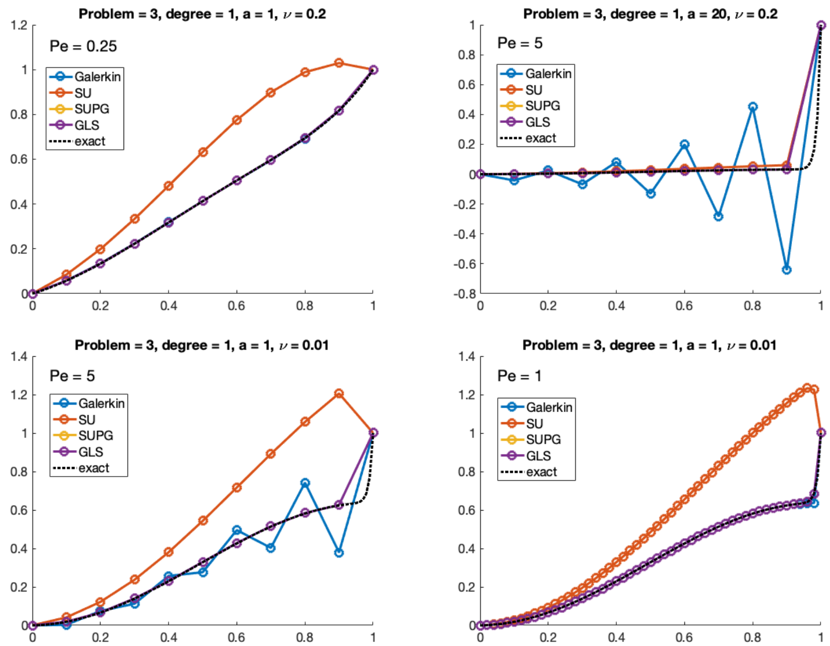

Figure 2: Problem 3 solved with linear elements

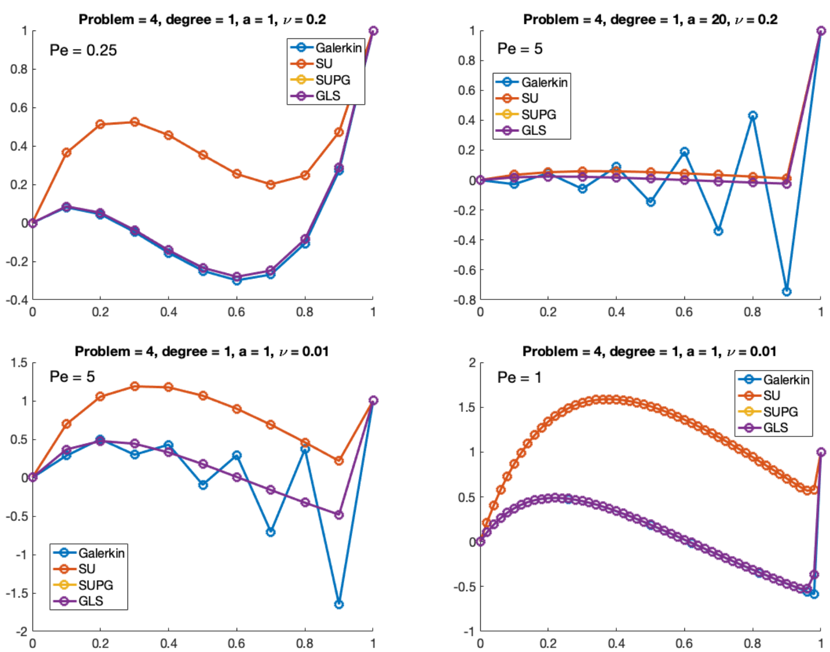

Figure 3: Problem 4 solved with linear elements

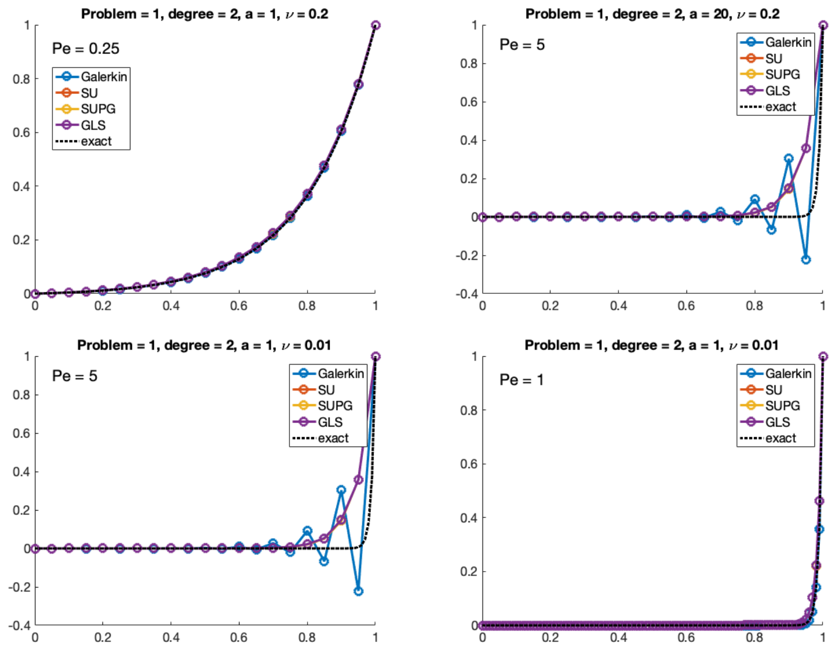

Figure 4: Problem 1 solved with quadratic elements

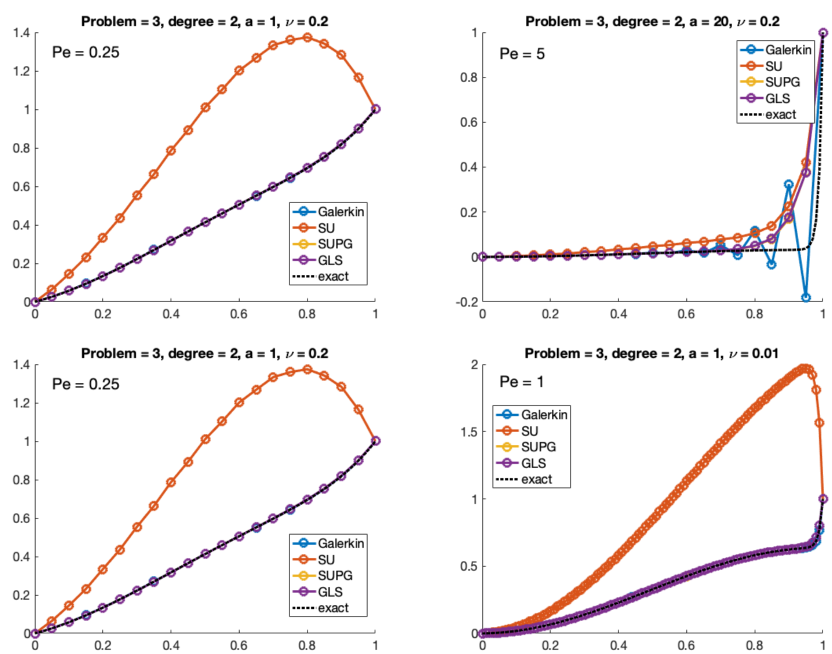

Figure 5: Problem 3 solved with quadratic elements

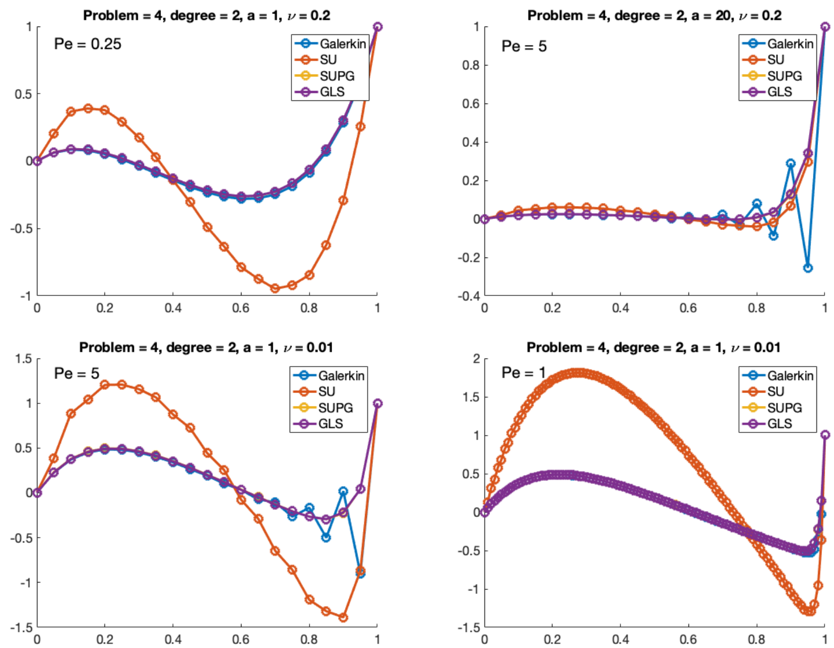

Figure 6: Problem 4 solved with quadratic elements

5 Codes

5.1 SU formulation

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


% reference element information

nen = referenceElement.nen;

ngaus = referenceElement.ngaus;

wgp = referenceElement.GaussWeigths;

N = referenceElement.N;

Nxi = referenceElement.Nxi;


% example properties

a = example.a;

nu = example.nu;

tau = example.tau;


% Number of nodes and elements
```

```
nPt = length(X);
nElem = size(T,1);

K = zeros(nPt,nPt);
f = zeros(nPt,1);

% Loop on elements
for ielem = 1:nElem
    Te = T(ielem,:);
    Xe = X(Te);
    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;
        w_ig = wgp(ig)*h/2;
        Ke = Ke + w_ig*(N_ig'*a*Nx_ig + Nx_ig'*nu*Nx_ig) ...
            + w_ig*(tau*a*Nx_ig)'*(a*Nx_ig);
        x = N_ig.*Xe; % x-coordinate of the gauss point
        s = SourceTerm(x,example);
        fe = fe + w_ig*s*(N_ig)';
    end
    % Assmebly
    K(Te,Te) = K(Te,Te) + Ke;
    f(Te) = f(Te) + fe;
end
```

5.2 SUPG formulation

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

```
% reference element information
nen = referenceElement.nen;
ngaus = referenceElement.ngaus;
wgp = referenceElement.GaussWeigths;
N = referenceElement.N;
Nxi = referenceElement.Nxi;
N2xi = referenceElement.N2xi;

% example properties
a = example.a;
nu = example.nu;
tau = example.tau;

% Number of nodes and elements
nPt = length(X);
nElem = size(T,1);

K = zeros(nPt,nPt);
f = zeros(nPt,1);

% Loop on elements
for ielem = 1:nElem
    Te = T(ielem,:);
    Xe = X(Te);
    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;
        N2x_ig = N2xi(ig,:)*2/h;
        w_ig = wgp(ig)*h/2;
        Ke = Ke + w_ig*(N_ig'*a*Nx_ig + (nu + tau* norm(a)^2)*Nx_ig'*Nx_ig);
```

```
        x = N_ig*Xe; % x-coordinate of the gauss point
        s = SourceTerm(x,example);
        fe = fe + w_ig*(N_ig)'*s ...
            + w_ig*(tau*a*Nx_ig)'*s;
    end
    % Assmebly
    K(Te,Te) = K(Te,Te) + Ke;
    f(Te) = f(Te) + fe;
end
```

5.3 GLS formulation

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

% reference element information
nen = referenceElement.nen;
ngaus = referenceElement.ngaus;
wgp = referenceElement.GaussWeigths;
N = referenceElement.N;
Nxi = referenceElement.Nxi;
N2xi = referenceElement.N2xi;

% example properties
a = example.a;
nu = example.nu;
tau = example.tau;

% Number of nodes and elements
nPt = length(X);
nElem = size(T,1);

K = zeros(nPt,nPt);
f = zeros(nPt,1);

% Loop on elements
```

```
for ielem = 1:nElem
    Te = T(ielem,:);
    Xe = X(Te);
    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;
        N2x_ig = N2xi(ig,:)*2/h;
        w_ig = wgp(ig)*h/2;
        Ke = Ke + w_ig*(N_ig'*a*Nx_ig +(nu + tau* norm(a)^2)*Nx_ig'*Nx_ig)...
            + w_ig*tau*nu*((a*Nx_ig)'*N2x_ig - N2x_ig'*(a*Nx_ig) + nu*N2x_ig'*N2x_ig);
        x = N_ig*Xe; % x-coordinate of the gauss point
        s = SourceTerm(x,example);
        fe = fe + w_ig*(N_ig)'*s ...
            + w_ig*(tau*a*Nx_ig)'*s...
            + w_ig*tau*nu*(N2x_ig)'*s;
    end
    % Assmebly
    K(Te,Te) = K(Te,Te) + Ke;
    f(Te) = f(Te) + fe;
end
```