MATLAB REPORT-1

- Anurag Bhattacharjee

<u>All formulations computed for p=1</u>

1) First example solved with Galerkin's method
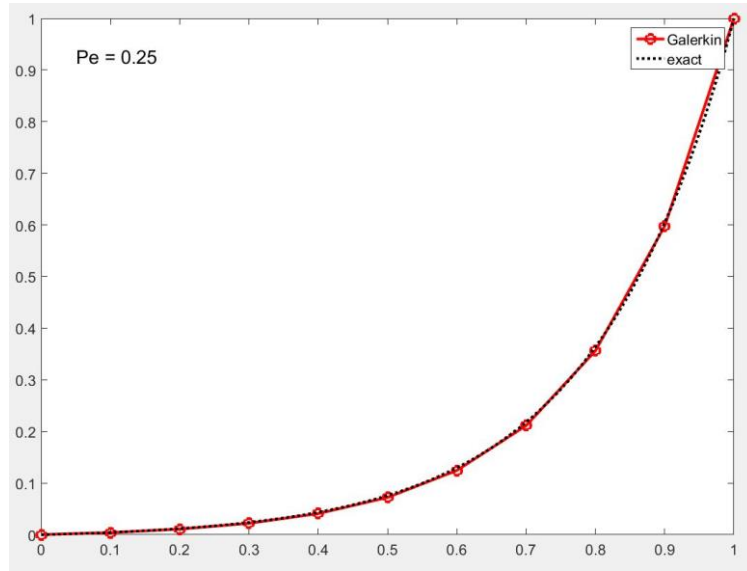<u>Case-1</u>

<u>Fig-1 Galerkin Solution for first case</u>

<u>Case-2 & 3</u>

       Since the Peclet number is less than 1, we get exact solution as is expected from the Galerkin formulation.
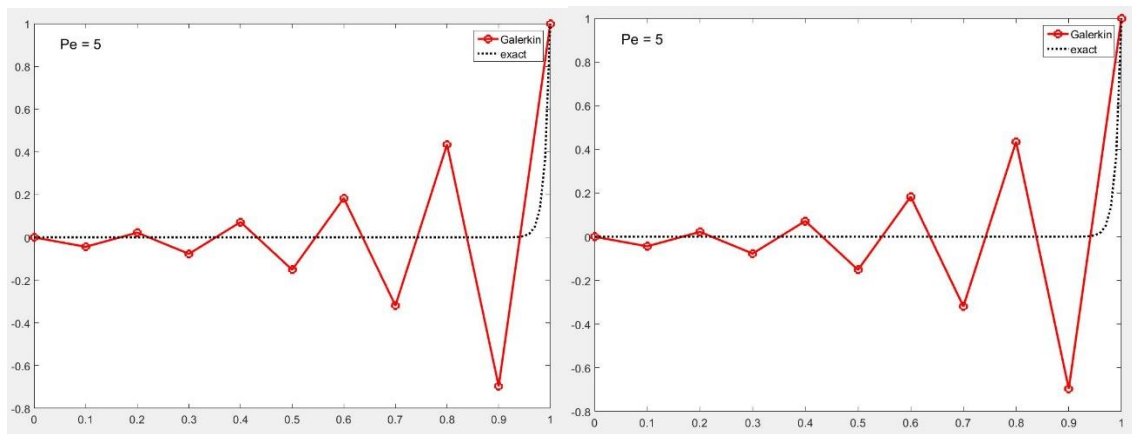

<u>Fig2- Left to right- Galerkin solution for second and third cases</u>

       For both these cases we observe that the Peclet number is 5 (Pe>1), so we get oscillations in the solution which is in accordance with the Galerkin formulation.
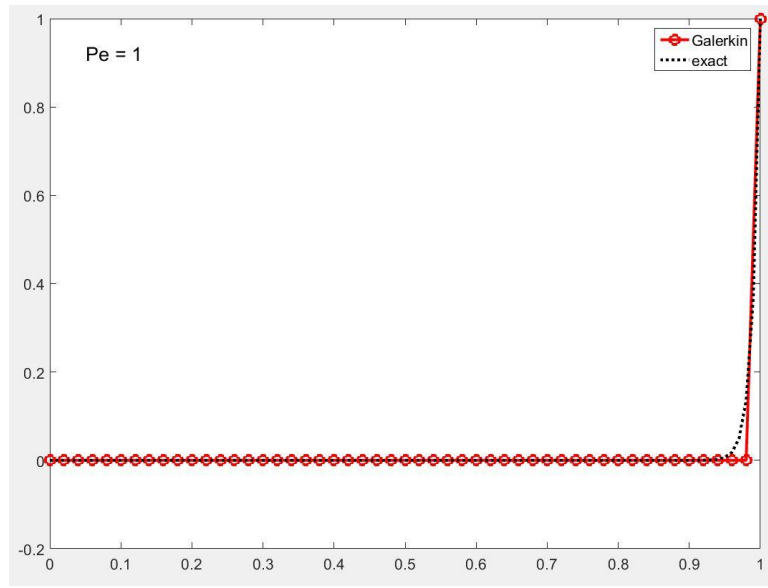
Fig3- Galerkin solution for fourth case

As the value of Peclet number in this case is not larger than one we get a stable solution.

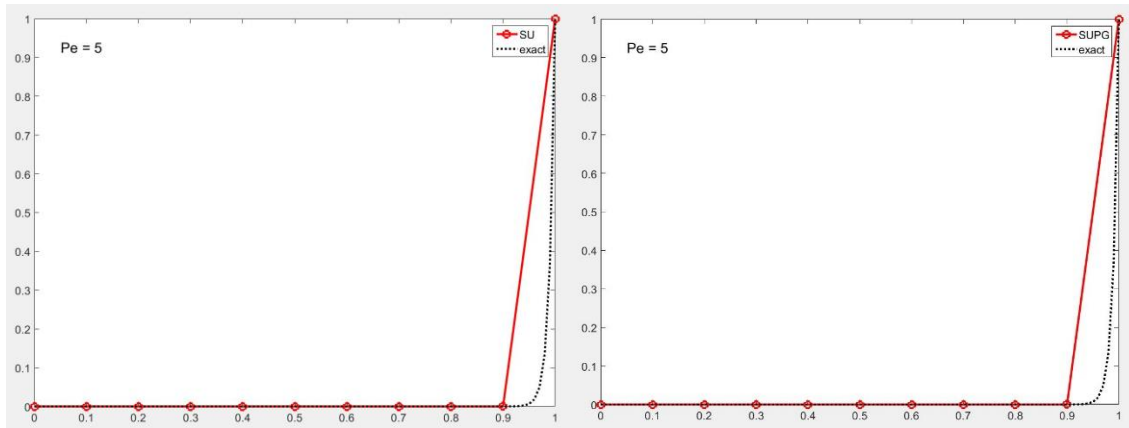**2)** Solution of first example using SU, SUPG and GLS formulations using the third case

Fig4- SU and SUPG solution

The SU, SUPG and GLS formulations all show stable results for the third case with Peclet number 5 as is expected with a constant source term.
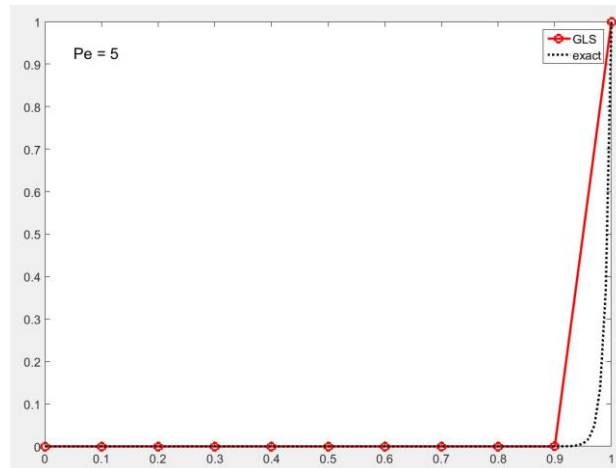

Fig5- GLS Solution

As we are computing all the formulations for 1-D Linear Element, the diffusion terms disappear from both the residual R(u) and the Þ(w). So the all the formulations except Galerkin yield the same solution.

The changes in the matlab code for the gauss loop for the different formulations can be seen below-

Galerkin Formulation

```
36              % Loop on Gauss points
37 -    ⊟       for ig = 1:ngaus
38 -                N_ig = N(ig,:);
39 -                Nx_ig = Nxi(ig,:)*2/h;
40 -                w_ig = wgp(ig)*h/2;
41 -                Ke = Ke + w_ig*(N_ig'*(a*Nx_ig) + Nx_ig'*(nu*Nx_ig));
42 -                x = N_ig*Xe; % x-coordinate of the gauss point
43 -                s = SourceTerm(x,example);
44 -                fe = fe + w_ig*(N_ig')*s;
45 -            end
```

SU Formulation

```
37              % Loop on Gauss points
38 -    ⊟       for ig = 1:ngaus
39 -                N_ig = N(ig,:);
40 -                Nx_ig = Nxi(ig,:)*2/h;
41 -                w_ig = wgp(ig)*h/2;
42 -                Ke = Ke + w_ig*(N_ig'*a*Nx_ig + Nx_ig'*nu*Nx_ig) ...
43                      + w_ig*(tau*a*Nx_ig)'*(a*Nx_ig);
44 -                x = N_ig*Xe; % x-coordinate of the gauss point
45 -                s = SourceTerm(x,example);
46 -                fe = fe + w_ig*(N_ig)'*s;
47 -            end
```

SUPG Formulation

```
37              % Loop on Gauss points
38 -    ⊟       for ig = 1:ngaus
39 -                N_ig = N(ig,:);
40 -                Nx_ig = Nxi(ig,:)*2/h;
41 -                w_ig = wgp(ig)*h/2;
42 -                Ke = Ke + w_ig*(N_ig'*(a*Nx_ig) + Nx_ig'*(nu*Nx_ig))+...
43                      w_ig*(tau*a*Nx_ig)'*(a*Nx_ig); %Stabilization term added
44 -                x = N_ig*Xe; % x-coordinate of the gauss point
45 -                s = SourceTerm(x,example);
46 -                fe = fe + w_ig*(N_ig')*s + w_ig*a*Nx_ig'*tau*s; %Source stabilization
47 -            end
```

GLS Formulation

```
37              % Loop on Gauss points
38 -    ⊟       for ig = 1:ngaus
39 -                N_ig = N(ig,:);
40 -                Nx_ig = Nxi(ig,:)*2/h;
41 -                w_ig = wgp(ig)*h/2;
42 -                Ke = Ke + w_ig*(N_ig'*(a*Nx_ig) + Nx_ig'*(nu*Nx_ig))+ ...
43                      + w_ig*(tau*a*Nx_ig)'*(a*Nx_ig);
44                  %Neglected the diffusion term because of 1-D linear element.
45 -                x = N_ig*Xe; % x-coordinate of the gauss point
46 -                s = SourceTerm(x,example);
47 -                fe = fe + w_ig*(N_ig')*s + w_ig*a*Nx_ig'*tau*s; %Source stabilization
48 -            end
```

As can be seen from the codes the stiffness matrix (Ke) for all the formulations are similar as the second order diffusion terms are dropped because the computations are done in a 1-D linear domain.

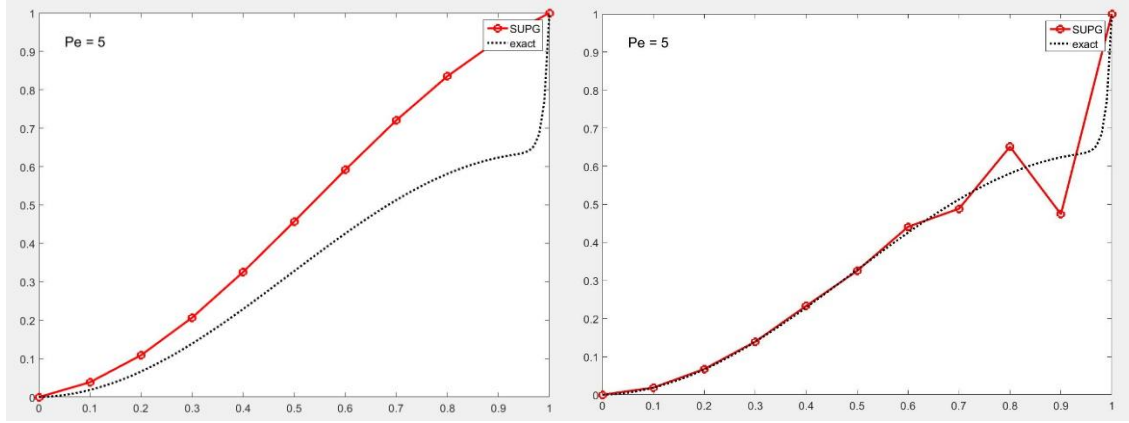**3)** Variation of SUPG solutions with variation of value of Tau

Fig6- Left to right- SUPG solutions for Tau=1 and Tau=0.01

Since the added stabilization parameter Tau is non-symmetric, higher values of Tau can lead to inconsistency in the solution as can be seen from the solution obtained for Tau= 1. Also, the higher value of Tau can be a result of a higher step size which can bring in some errors in computation. Some inconsistency can also be observed for Tau=0.01 which can be attributed to the sharp convergence to the boundary value of the exact solution.
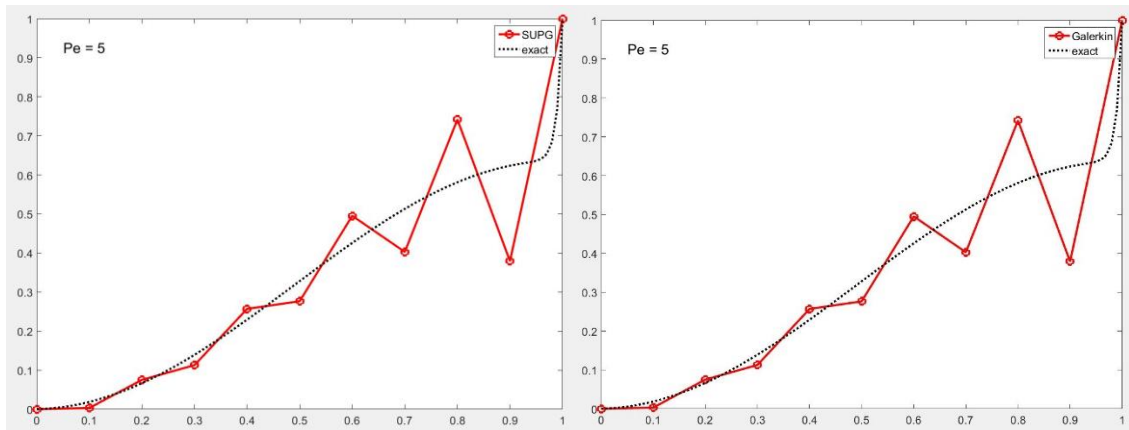

Fig7- Left to Right- SUPG solution for Tau= 0 and Galerkin solution for third example

When Tau =0, it can be observed that the SUPG solution behaves the same way as Galerkin solution. This is because as Tau= 0, the extra stabilization term in SUPG formulation [P(w)*Tau*R(u)] goes to zero and the formulation becomes same as Galerkin formulation.

4) Solutions for different formulations for the third problem using default values of parameters
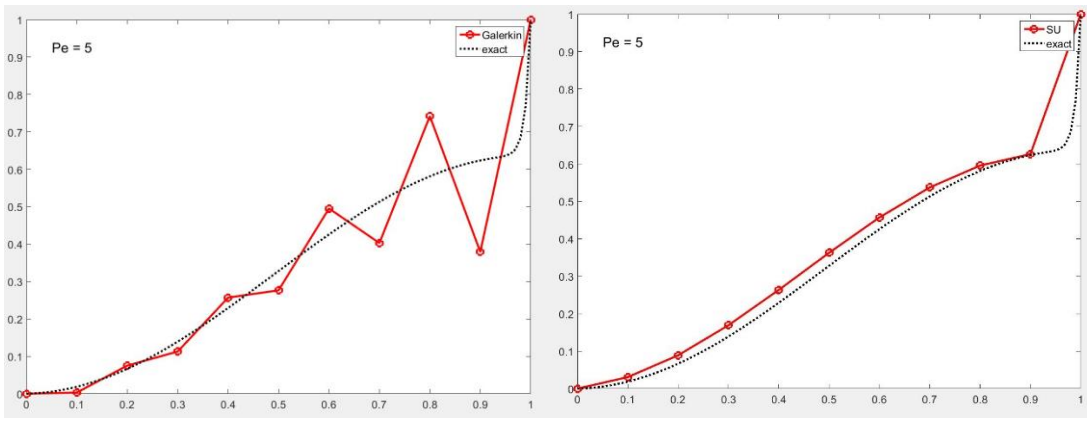

Fig8- Left to right- Galerkin solution and SU solution for third problem

Since the Peclet number is (5>1), the Galerkin formulation breaks down and the solution appears to be oscillating about the exact solution. The SU formulation shows a small deviation from the exact solution as the SU formulation is only effective for solutions with either constant or zero values. As the third problem has a variable solution, the SU formulation breaks down.
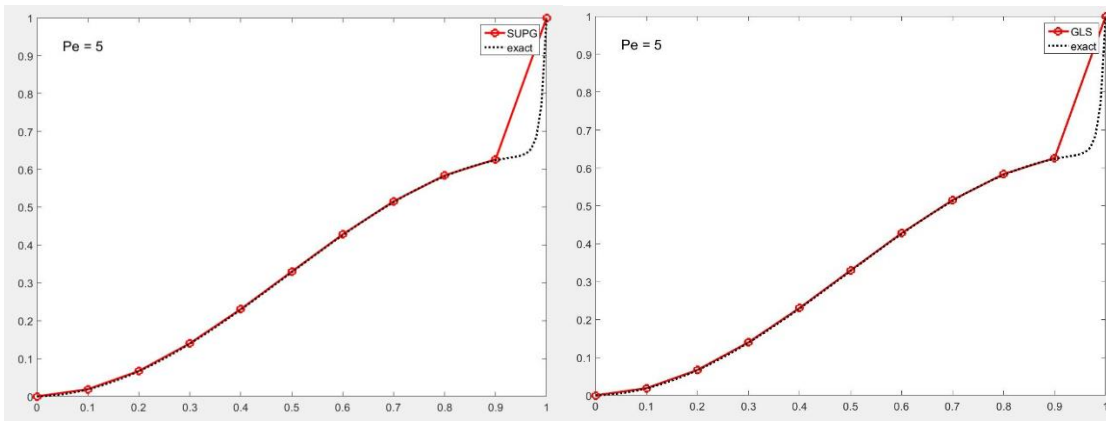

Fig9- Left to right- SUPG formulation and GLS formulation for third problem

Unlike the SU formulation, both the SUPG and GLS formulations have stabilizations for the solution terms that account for variable solutions. Therefore, we get consistent solutions for both these formulations.