Convection Diffusion

Doremus Alexandre

June 2019

Table of contents

1 Introduction

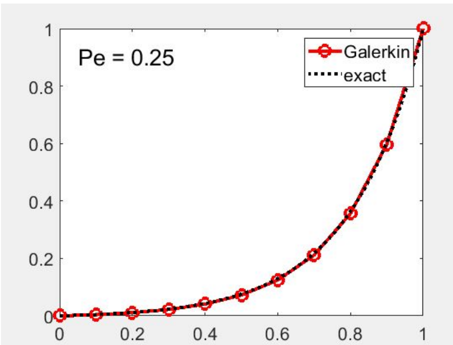
The problem to be solved is the following steady convection-diffusion problem :

$$\begin{cases} au_x - \nu u_{xx} &= s, \quad x \in [0,1] \\ u(0) &= u_0 \\ u(1) &= u_1 \end{cases} \tag{1}$$

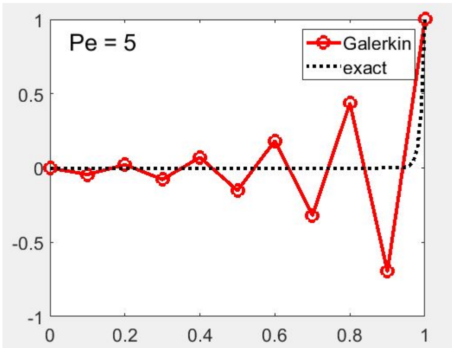
2 Question1

We will solve in this part with the following numerical values for the parameters :

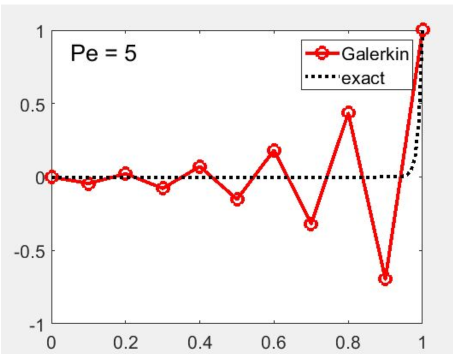
$$\begin{cases} s &= 0 \\ u(0) &= 0 \\ u(1) &= 1 \end{cases} \tag{2}$$



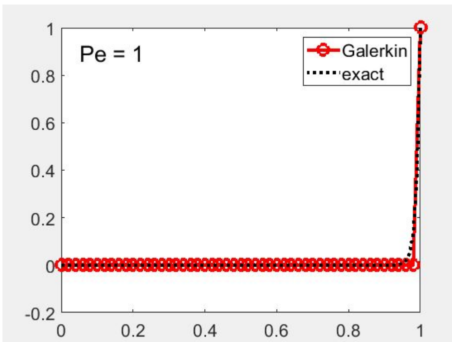
(a) a = 1, ν = 0.2, 10 linear elements



(b) a = 20, ν = 0.2, 10 linear elements



(a) a = 1, ν = 0.01, 10 linear elements



(b) a = 1, ν = 0.01, 50 linear elements

We can see that as according to the theory developped in class, when the Peclet number is smaller or equal to one we obtain the exact solution using Galerkine but then with P_e>1, oscillations appear.

3 Question 2

3.1 Comparison of the three methods : SU, SUPG and GLS

We will solve in this part with the following numerical values for the parameters :
a = 1
ν= 0.01
10 linear elements

3.1.1 SU

The results are shown in the following graph :

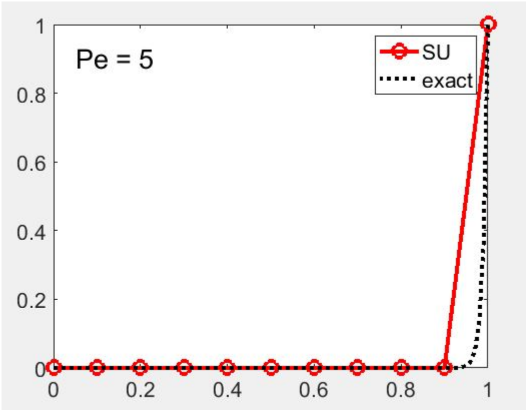


FIGURE 1 – SU

3.1.2 SUPG

As the only method already coded is Galerkine, what we need is to add some terms in the discretization part of the code to obtain the SUPG and GLS formulations.
For SUPG we need to add this term to K_e : + $w_i g * (tau * a * Nx_i g)' * (a * Nx_i g) - w_i g * (tau * a * Nx_i g') * (nu * N2x_i g)$
and this term to f_e : $+w_i g * (tau * a * Nx_i g') * s$

The results are shown in the following graph :

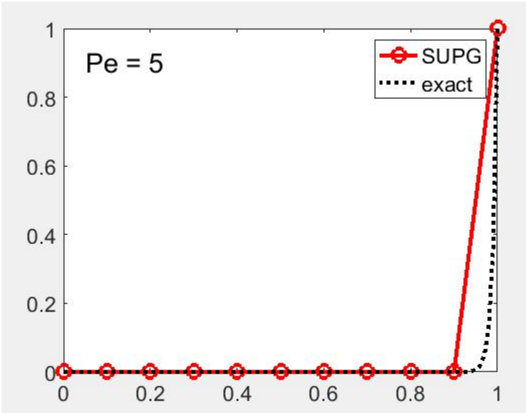


FIGURE 2 – SUPG

3.1.3 GLS

For GLS we need to add this term to K_e : $+\ w_i g * (tau * a * Nx_i g)' * (a * Nx_i g) - wig * (tau * a * Nx_i g') * (nu * N2x_i g) - w_i g * (tau * nu * N2x_i g') * (a * Nx_i g) + wig * (tau * nu * N2x_i g') * (nu * N2x_i g)$
and this term to f_e : $+w_i g * (tau * a * Nx_i g') * s - w_i g * (tau * nu * N2x_i g') *) * s$

The results are shown in the following graph :

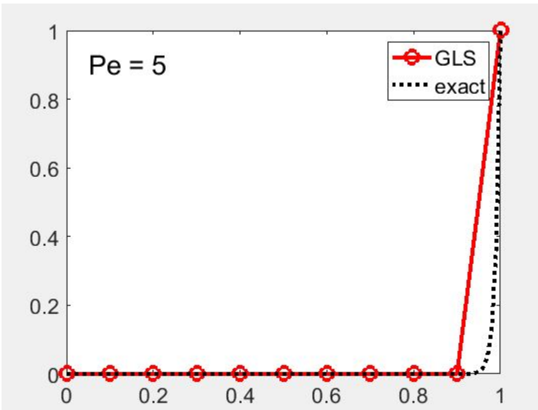


FIGURE 3 – GLS

Do we obtain the exact solution ?

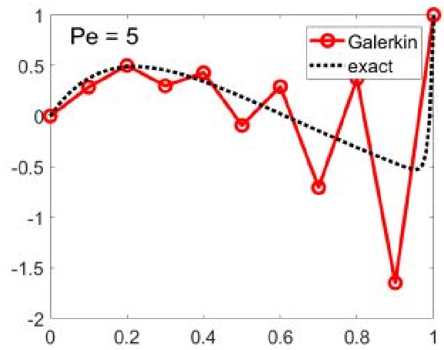
We do not obtain the exact solution with any of the previous method. What's more, we can see that the numerical solution is the same for SU, SUPG and GLS with these parameters. It could look like there is no need to have so much methods, but it is due to the fact that with linear elements the added terms are in the end equal due to the null second derivatives.

3.2 New exponential source term

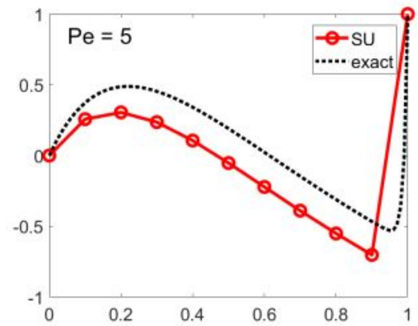
We will solve in this part with the following numerical values for the parameters :

$$\begin{cases} s &= 10e^{-5x} - 4e^{-x} \\ u(0) &= 0 \\ u(1) &= 1 \end{cases} \tag{3}$$

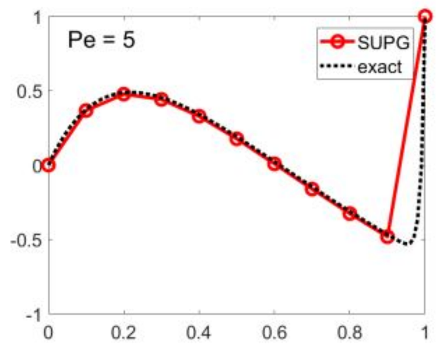
We obtain the following curves :



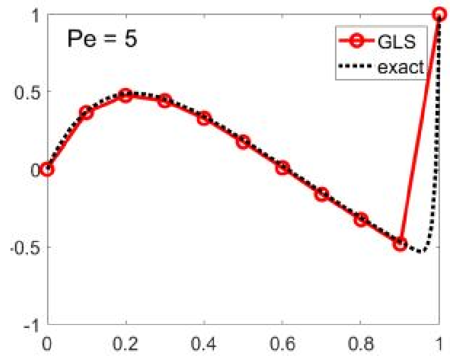
(a) Galerkine



(b) SU



(a) SUPG



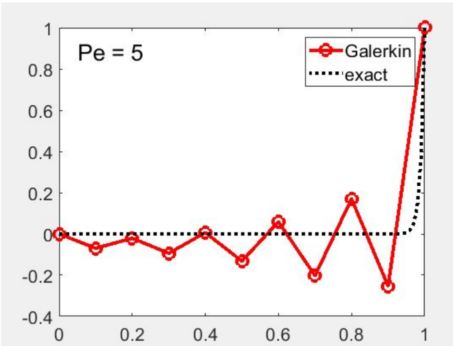
(b) GLS

The Galerkine method still shows big oscillations because of a too large Péclet number. Additionally, we can see that this time SU doesn't perform as well. This was to be expected because SU's formulation is not consistent, therefore it doesn't perform well for non constant terms. On the other hand, SUPG and GLS perform better, which is logic since they were designed to be consistent and hence perform better with this type of source term.

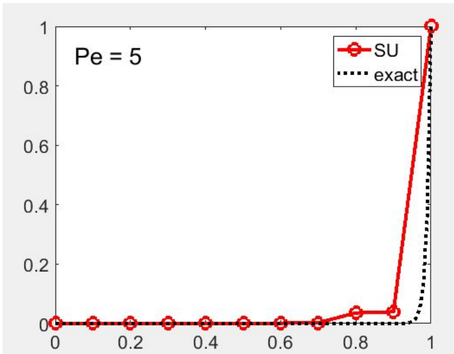
3.3 Quadratic elements

For the implementation of quadratic elements we first need to define the table of connectivity T for quadratic elements, which we can do as follows :
if p==1
T = [1 :nPt-1 ; 2 :nPt]' ;
elseif p ==2
T = [1 :nPt-2 ; 2 :nPt-1 ; 3 :nPt]' ;

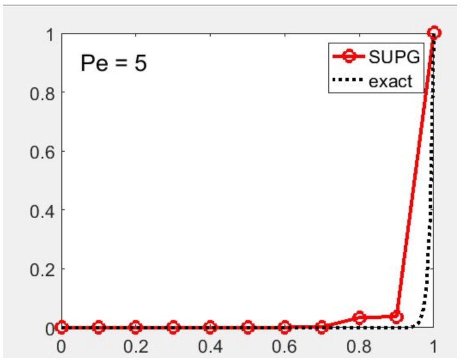
What's more, some second order derivatives would appear in the discretization part and we would need new functions for them which we will call *N2xi* :
N2xi = referenceElement.N2xi ;
*N2x ig = N2xi(ig, :)*2/h ;*



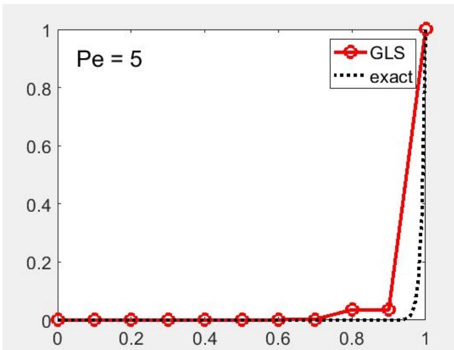
(a) Galerkine



(b) SU



(a) SUPG



(b) GLS

If we compare with 1D, the oscillations for Galerkine are reduced. As for SU, SUPG and GLS, our numerical solution is no longer nodally exact, which is due to the parameter tau which ought to be different in 2D. By zooming in, we also see that while close, the SU solution performs slightly worse than the others as it tends to be further from the solution faster in the ascending phase of the curve.